

The Effects of Target Location Uncertainty in Game Theoretic Solutions to Optimal Trajectory Formulations

by

Daniel Mark Morales

B.S., United States Air Force Academy (2003)

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of

Master of Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2005

© 2005 Massachusetts Institute of Technology. All rights reserved.

The author hereby grants to Massachusetts Institute of Technology permission to
reproduce and
to distribute copies of this thesis document in whole or in part.

Signature of Author

Department of Aeronautics and Astronautics
6 May 2005

Certified by

Randy Avent
Leader, Group 61
MIT Lincoln Laboratory
Thesis Supervisor

Certified by

Charles Coleman
Boeing Assistant Professor of Aeronautics and Astronautics
Thesis Supervisor

Accepted by

Jaime Peraire
Professor of Aeronautics and Astronautics
Chair, Committee on Graduate Students

Report Documentation Page			Form Approved OMB No. 0704-0188	
<p>Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.</p>				
1. REPORT DATE 00 JUN 2005	2. REPORT TYPE N/A	3. DATES COVERED -		
4. TITLE AND SUBTITLE The Effects of Target Location Uncertainty in Game Theoretic Solutions to Optimal Trajectory Formulations			5a. CONTRACT NUMBER	
			5b. GRANT NUMBER	
			5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S)			5d. PROJECT NUMBER	
			5e. TASK NUMBER	
			5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Massachusetts Institute Of Technology			8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)			10. SPONSOR/MONITOR'S ACRONYM(S)	
			11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release, distribution unlimited				
13. SUPPLEMENTARY NOTES The original document contains color images.				
14. ABSTRACT				
15. SUBJECT TERMS				
16. SECURITY CLASSIFICATION OF: a. REPORT b. ABSTRACT c. THIS PAGE unclassified unclassified unclassified			17. LIMITATION OF ABSTRACT UU	18. NUMBER OF PAGES 233
				19a. NAME OF RESPONSIBLE PERSON

**The Effects of Target Location Uncertainty in Game Theoretic Solutions to
Optimal Trajectory Formulations**

by

Daniel Mark Morales

Submitted to the Department of Aeronautics and Astronautics
on 6 May 2005, in partial fulfillment of the
requirements for the degree of
Master of Science

Abstract

The integration of a variety of Intelligence, Surveillance, and Reconnaissance (ISR) assets is vital to the acquisition of knowledge critical to battlefield success. Missions performed by a penetrating Unmanned Aerial Vehicle (UAV) are of particular interest. The environment in which UAVs must operate includes Surface-to-Air Missile (SAM) sites with extended ranges, among other threats. SAM location uncertainty, terrain obscuration, and radar/sensor capabilities all contribute to the complexity of the situation.

This thesis provides a game theoretic approach to determine optimal UAV strategies against enemy SAM sites. It is shown that most characteristics of the UAV or SAM have negligible effects on both image quality I_q and probability of kill P_k (probability of the SAM shooting down the UAV). Instead, SAM location uncertainty has the largest influence. After only 0.5 miles of uncertainty, the P_k of a UAV assuming perfect knowledge of the SAM location rises to 0.56. When the uncertainty rises to about four miles, the P_k rises to 0.99.

When the UAV takes uncertainty into account, the results are not much better. Assuming that the SAM may be at one of three possible locations, the result is an average P_k of 0.49 or 0.79, depending on which optimization routine was used. Extending this situation to five, seven, and nine possible SAM locations results in an increase in P_k to 0.99 at seven locations. An even more realistic scenario involving the UAV optimizing a path through a large area of varying probabilities results in an 85% chance of getting shot down if the SAM is located within a five mile radius of the center of the area. Outside of this area, the UAV is guaranteed to get shot down with a P_k of 0.99. Other techniques and methods must be explored and used in combination with Radar Cross Section (RCS) management to ensure the continuing collection of valuable ISR imagery in the coming years.

Thesis Supervisor: Randy Avent
Title: Leader, Group 61
MIT Lincoln Laboratory

Thesis Supervisor: Charles Coleman
Title: Boeing Assistant Professor of Aeronautics and Astronautics

Acknowledgements

First and foremost, I would like to thank my advisor, Randy Avent, who gave his time selflessly and guided me well with his knowledge and wisdom. Les Servi, who wrote Appendix F.3, also served as an invaluable resource of whom I took advantage quite often. Others who have aided me in the writing of this thesis include Linda Zeger, Mike Pietrucha, Dave Genovese, John Kuconis, Rich Trentman, Kevin Kelly, Steven Lee, Larry Bush, Chuck Burt, Rajesh Viswanathan, Nagabushan Sivananjaiah, Jason Oda, Mike Gruber, Mike Spitalere, Art Spear, Hal Heggestad, Michael de la Maza, Peter Mastromarino, Jonathan Kurz, Michael Toups, L. Keith Sisterson, Mary Carey, and Lisa Quinn. This list does not include the countless people who have offered me words of encouragement throughout my time at MIT, which is just as important. Finally, I would like to thank my family, my friends, and my God for their love and support.

Disclaimer

The views expressed in this article are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or U.S. Government.

Contents

1	Introduction	12
1.1	SAMs	13
1.2	Human Analysis	13
1.3	Automatic Target Recognition	14
1.4	Cueing Strategy	16
1.5	Penetrating UAVs	21
1.6	Scenarios	21
2	UAV Strategies	24
2.1	Combinatorial Problems	24
2.2	Typical Aircraft Tactics	26
2.3	UAV Strategy Formulation	28
2.3.1	Turn Constraints	29
2.3.2	Line of Sight Obscuration	30
2.3.3	Terrain Limitations	30
2.4	Image Quality	31
3	SAM Strategies	37
3.1	Detections	37
3.2	Probability of Kill	47
4	Game Theory	50
4.1	A Brief History of Game Theory	51

4.2	Characteristics of Game Theory	51
4.2.1	Nomenclature	52
4.3	Calculating the Nash Equilibrium	53
4.4	UAV vs SAM	54
5	Results and Conclusions	58
5.1	Case One	59
5.2	Case Two	63
5.3	Case Three	69
5.4	Case Four	74
5.5	Conclusions	75
5.6	Future Research	77
A	Surface to Air Missiles	84
A.1	Radar Development	84
A.2	SAM Development	86
A.3	Modern SAMs	92
B	Unmanned Aerial Vehicles (UAVs)	95
B.1	History of UAVs	95
B.1.1	Early History	96
B.1.2	World War II	96
B.1.3	Cold War	97
B.1.4	Post-Vietnam Development	100
B.2	UAVs Today	103
B.2.1	Types of UAVs	103
B.2.2	Control Methods	106
B.2.3	Why UAVs?	108
B.2.4	UAV Roles	109
B.2.5	Current Operations and Future Growth	110

C Radar Systems	113
C.1 Characteristics of Radar	113
C.1.1 Radar Wave Basics	114
C.1.2 The Doppler Shift	117
C.1.3 Radar Block Diagram	118
C.2 Radar Cross Section	121
C.2.1 Simple Targets	121
C.2.2 Complex Targets	124
C.2.3 Calculating RCS	128
C.2.4 Fluctuations	131
C.3 The Radar Equation	132
D Sensors and Image Quality	137
D.1 Sensors	137
D.2 Image Quality	142
D.2.1 National Imagery Interpretability Rating Scale (NIIRS)	142
D.2.2 General Image Quality Equation (GIQE)	147
E National Image and Interpretability Rating Scales (NIIRS)	150
E.1 Visible NIIRS March 1994	150
E.2 Radar NIIRS April 1999	153
E.3 Infrared NIIRS April 1996	157
E.4 Multispectral NIIRS January 2001	161
F Game Theory	165
F.1 A Brief History of Game Theory	166
F.2 Characteristics of Game Theory	166
F.2.1 Definition, Application, and Rationality	166
F.2.2 The Prisoners' Dilemma	167
F.2.3 Nomenclature	169
F.3 Calculating the Nash Equilibrium	170

G Program Code	173
G.1 CreateGraph.m	173
G.2 UAVvsSAM.m	181
G.3 DtedData.m	186
G.4 SetUpGrid.m	187
G.5 AngleDistance.m	189
G.6 gc2ecr.m	193
G.7 CalculateDistances.m	194
G.8 CreatePaths.m	196
G.9 Infeasible.m	206
G.10 Detect.m	210
G.11 FindPk.m	215
G.12 ImageQuality.m	218
G.13 FindA.m	220
G.14 Nash.m	221
G.15 Roll.m	223
G.16 ecr2gc.m	225
G.17 Visualize.m	227
G.18 create_uav.m	229
G.19 loop3d.m	231
G.20 matrix_interp.m	232

List of Figures

1-1	Typical ROC Curve	15
1-2	Probability of Successful Identification	17
1-3	ISR Platform Coverage Comparisons.	18
1-4	Probability of Blockage for Global Hawk UAV	19
1-5	Probability of Blockage for Predator UAV	20
1-6	Block Diagram of Main Process	22
2-1	UAV Strategy Creation	25
2-2	Doppler Shift	27
2-3	Possible Types of UAV Paths	28
2-4	Terrain Map of North Korea	31
2-5	Block Diagram of Image Quality Calculations	34
2-6	Color Map of Image Quality with Varying Azimuth and Depression	36
3-1	Block Diagram of Detection Process	38
3-2	UAV Model	39
3-3	RCS of UAV Model with Varying Aspect Angle	40
3-4	RCS of UAV Model as Frequency Varies	42
3-5	Color Map of Model RCS with Varying Depression and Azimuth	43
3-6	Sample UAV Trajectory	45
3-7	Example of Detection Cutoff	46
3-8	Block Diagram of Kill Probability Formulation	48
4-1	UAV Path given by Nash Equilibrium	56

5-1	Image Quality as Detection Radius Varies	61
5-2	Probability of Kill with Varying UAV Speed	62
5-3	Probability of Kill with Zero to 10 miles of Uncertainty	65
5-4	NIIRS Level with Zero to 10 mile Uncertainty	66
5-5	Probability of Kill with Zero to 100 mile Uncertainty	67
5-6	SAM Engagement and Location Uncertainty Areas	68
5-7	NIIRS Level with Zero to 100 mile Uncertainty	70
5-8	Probability of Kill as Possible SAM Locations Increase	73
5-9	Scenario Depiction for Case Four	74
A-1	RSC-51	87
A-2	SA-1 "Guild"	88
A-3	SA-2 Guideline	89
A-4	SA-6 Gainful	91
A-5	SA-10 Grumble	93
A-6	SA-20 Triumf	94
B-1	History of UAV Development	97
B-2	Lockheed D-21 UAV	98
B-3	Ryan BQM-34 Firebee	99
B-4	RQ-4A Global Hawk	101
B-5	GA RQ-1A Predator	102
B-6	Classes of UAVs	104
B-7	Ten Levels of Automation	107
B-8	Levels of Interaction	108
B-9	UAVs in Operation Iraqi Freedom	111
B-10	DoD UAV/UCAV Funding	112
C-1	Electromagnetic Spectrum	115
C-2	Pulse-Modulated Waveform Example	116
C-3	Envelope of Radar Receiver Output vs Time	117
C-4	Plot of Doppler Frequency vs Radial Velocity	119

C-5	Block Diagram of Radar	120
C-6	Approximate RCS Values	122
C-7	RCS of a Sphere	123
C-8	Sphere Backscatter from Short Pulse Radar	125
C-9	Components of Target RCS	126
C-10	T-33 Shooting Star	127
C-11	T-33 RCS Pattern	128
C-12	B-26 RCS Polar Plot	129
C-13	Measured and Calculated RCS of JGAM	130
C-14	Sources of Radar Noise	134
D-1	Wescam 14 EO/IR Sensor	138
D-2	Predator Video and IR Capability	139
D-3	Lynx SAR/GMTI	140
D-4	Predator SAR Capability	141
D-5	NIIRS Level 2 Image	144
D-6	NIIRS Level 6 Image	145
D-7	NIIRS Level 8 Image	146
D-8	Predicted and Observed NIIRS Values	149

List of Tables

1.1	Number of Image Analysts Needed for Various Countries	13
1.2	Number of Image Analysts Needed To Resolve False Alarms	14
2.1	Best and Worst Parameter Values	35
5.1	Input Parameter Values for Case One Test Cases	59
5.2	SAM Locations for Scenario One in Case Three	71
5.3	Case Three Results for Scenario One using Average Payoffs	71
5.4	Case Three Results for Scenario One using Minimum Payoffs	72
5.5	SAM Locations for Scenario Two in Case Three	73
5.6	Case Four Results	75
F.1	Prisoners' Dilemma	168

Chapter 1

Introduction

When it becomes necessary to plan or engage in armed combat with an adversary, persistent and accurate knowledge of the enemy is perhaps the most important tool available to military commanders. That knowledge provides a plethora of critical information, such as enemy air defense locations, troop strength, or command structure, which allows friendly forces the opportunity to adapt and react. The seamless integration of a variety of ground, air, and space assets is vital to the acquisition of that knowledge and to the success of a particular battle or war. This thesis focuses on Intelligence, Surveillance, and Reconnaissance (ISR) missions performed by a penetrating Unmanned Aerial Vehicle (UAV).

The environment in which UAVs must operate during combat is dynamic and dangerous. Surface-to-Air Missile (SAM) sites with extended ranges and advanced missile capabilities pose a dire threat to penetrating aircraft. The locations of individual SAM sites may be uncertain or unknown. Terrain obscuration and radar/sensor capabilities also play a large factor in determining the outcome of such engagements. To that end, the goals of this thesis are the following:

- To investigate the use of a game-theoretic approach in determining optimal UAV strategies against enemy SAM sites.
- To model the radar detection and image acquisition processes for this specific combat situation.
- To determine the feasibility of an algorithmic approach to UAV path planning.

	Area (km²)	IAs Needed
North Korea	120,540	2,143
Iraq	437,072	7,771
Afghanistan	647,500	11,512
China	9,596,960	170,613

Table 1.1: Number of Image Analysts Needed for Various Countries

1.1 SAMs

SAMs have become increasingly more sophisticated. The SA-6 "Gainful" was introduced in the 1960s and can detect aircraft up to 150 km away [34]. The SA-10 "Grumble," introduced in 1980, was designed to shoot down low altitude, high speed aircraft or missiles and can detect and engage targets at a range of over 200 km [34]. Most recently, the SA-20 "Triumf," scheduled to deploy in 2005, can engage targets as far as 400 km away [34]. This type of advanced SAM can severely hinder the ability of aircraft to obtain precise, high resolution imagery. A more extensive discussion of the history and development of SAMs is located in Appendix A.

Before the start of an engagement, the probable locations of different SAMs are often known. However, once the battle commences, the SAMs are usually moved quickly and become very difficult to locate. The resulting problem then becomes finding SAMs that are dispersed throughout a region.

1.2 Human Analysis

One way to find the SAM sites is to use human image analysts (IAs) to physically scrutinize every image taken over the entire region by satellites or other assets with a wide field of view (FOV). However, it takes on average 30 minutes for eight IAs to identify targets in a 25 km² image [52]. Using the fact that SAMs typically move every nine hours, finding every SAM location using this method is impossible. Table 1.1 illustrates this point. A country like China with an area of almost 10 million km² would require over 170,000 image analysts to fully search the entire region. Even a smaller country like North Korea would require over 2,000 IAs. Clearly, another method must be found.

	Area (km²)	IAs Needed for ATR
North Korea	120,540	2,462
Iraq	437,072	8,924
Afghanistan	647,500	13,220
China	9,596,960	195,938

Table 1.2: Number of Image Analysts Needed To Resolve False Alarms

1.3 Automatic Target Recognition

Software that attempts to automatically recognize certain targets is called automatic target recognition (ATR). ATR utilizes a library of images of different targets taken at certain depression angles with varying azimuth angles. When given a new image, ATR algorithms will calculate the probability of finding a certain target according to a Gaussian probability distribution. It will then minimize the mean square error, which is a measure of the differences between the new image and the target library. This is equivalent to finding the maximum of the probabilities, which then gives the resulting identification. However, a minimum probability threshold must be set for detection of the target. This threshold causes a certain amount of false alarms to be generated. A false alarm occurs when a target is reported to be identified but does not actually exist.

A typical receiver operating characteristics (ROC) curve is shown in Figure 1-1 with the bottom axis on a logarithmic scale. As the probability of detecting the target increases, the number of false alarms per square kilometer increases as well. If it were assumed that a 90% probability of detection is sufficient, then a false alarm rate of 4.9 per square kilometer must be processed accordingly. On average, it takes IAs two minutes to resolve a false alarm. Referring back to the countries in Table 1.1, it is clear that this false alarm rate would overwhelm available resources. The specific numbers of IAs needed in this case are detailed in Table 1.2. Searching North Korea for possible SAM locations would still require over 2,000 IAs. Thus, ATR does not significantly reduce the demand for IAs.

One way to decrease the number of IAs needed is to reduce the search area by performing terrain delimitation. Approximately 40% of North Korea is not trafficable, meaning it would be impossible to place a SAM in those areas. This leaves only 60% of the country to be searched. However, that only reduces the number of IAs needed by 985, leaving a requirement of well

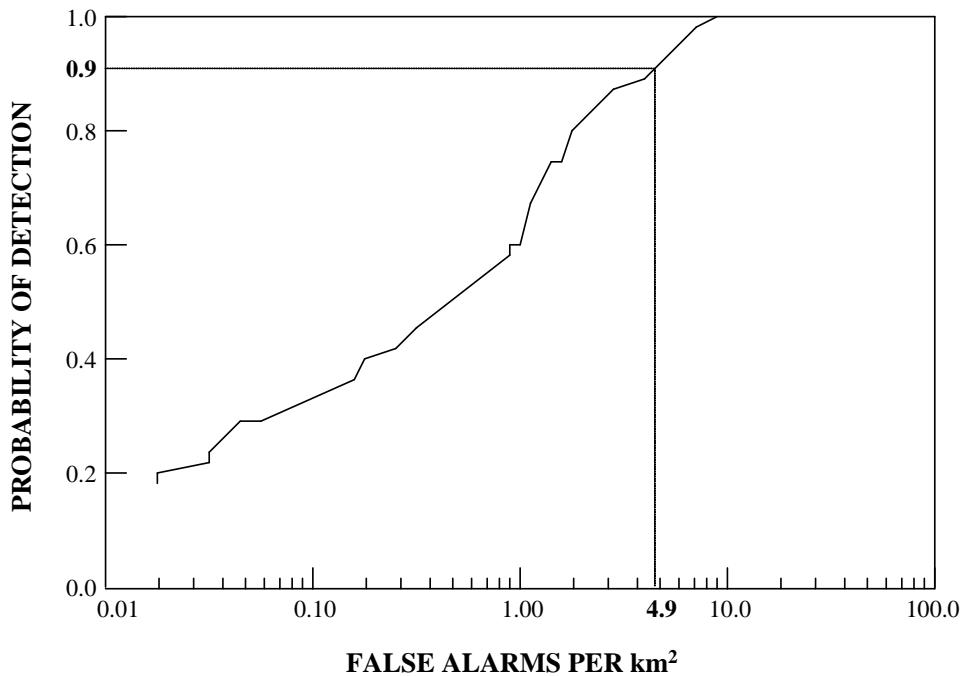


Figure 1-1: Typical ROC Curve. The probability of detecting the target is on the left side of the graph, and the the number of false alarms per square kilometer is on the bottom axis on a logarithmic scale. If a 90% probability of detection is desired, then a false alarm rate of 4.9 per square kilometer must be processed accordingly.

over 1,000 IAs. Although more sophisticated ATR algorithms could decrease the false alarm rate, the practical impact on the number of IAs needed would be insignificant.

1.4 Cueing Strategy

The logical extension of this reasoning leads to multiple sensors. One popular implementation involves using a wide FOV sensor cueing a narrow FOV sensor. The wide FOV sensor must be utilized with a low detection threshold, producing many false alarms. It would be a passive, satellite system that can search large areas. After identifying an area of interest, the wide FOV sensor will cue a different sensor with higher resolution capabilities. This sensor will have a narrower FOV and will perform single-class identification because it already knows the target is somewhere within the cued area. Figure 1-2 demonstrates the high probability of successful identification for a narrow FOV sensor searching a small area. Given those results, a strategy involving wide FOV sensors cueing narrow FOV sensors makes sense. However, due to the limited supply of narrow FOV satellites, a high resolution image could incur a time penalty of up to several days as the satellite must position itself appropriately. Since SAMs will move on average every nine hours, the belated satellite image would be inconsequential. Thus, the narrow FOV sensor must be an airborne asset.

A comparison of various sensor platforms and their associated FOVs is shown in Figure 1-3. A satellite system such as the future Space Based Radar (SBR) constellation has a very large FOV whereas the E-8 JSTARS aircraft has a much smaller FOV and high resolution. However, airborne assets such as the E-8 are not well suited to this particular mission. They would be unable to penetrate deep inside enemy territory due to their vulnerability to the SAM threat. They could remain a certain distance away from an area of operations, but that would severely hinder their ability to gather ISR data on targets deep within enemy territory. At that distance, terrain obscuration becomes a serious problem. Figures 1-4 and 1-5 depict the probability of blockage (PB) in Iraq and North Korea at altitudes of 65,000 ft and 15,000 ft, which represent typical operating altitudes of the RQ-4 Global Hawk and RQ-1 Predator UAVs, respectively. Although Iraq's flat desert areas imply low values of PB, the PB for the Global Hawk rises to almost 50% at a distance of 100 miles in North Korea, and the PB for the Predator is even

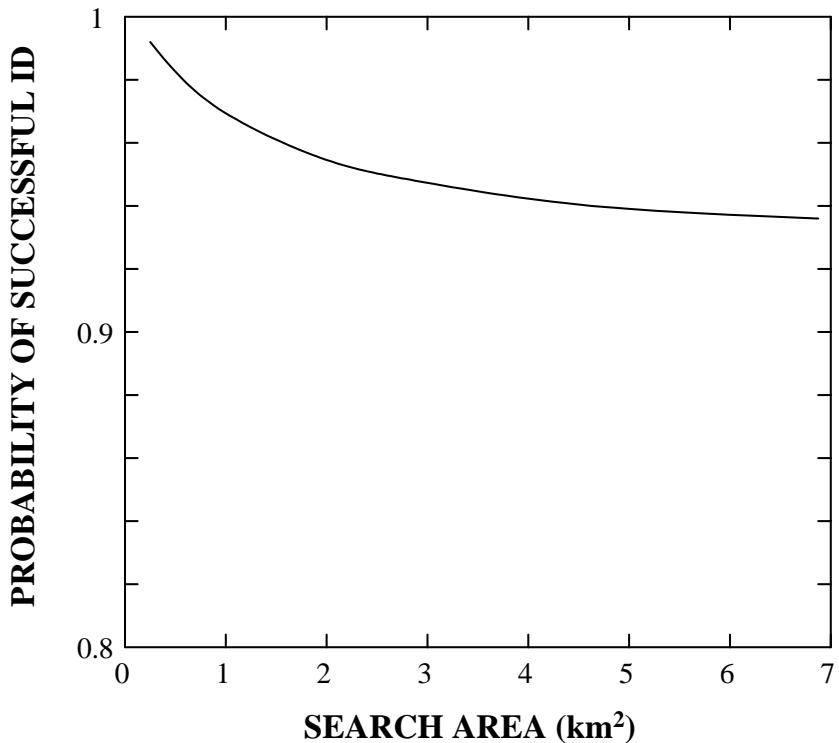


Figure 1-2: Probability of Successful Identification. This value on the left axis decreases only slightly as the search area gets larger on the bottom axis. These results show that a narrow FOV sensor can more easily identify a target in a small search area. If that step is a component in a cueing strategy involving multiple sensors, then numerous identifications could be made without requiring large numbers of image analysts.

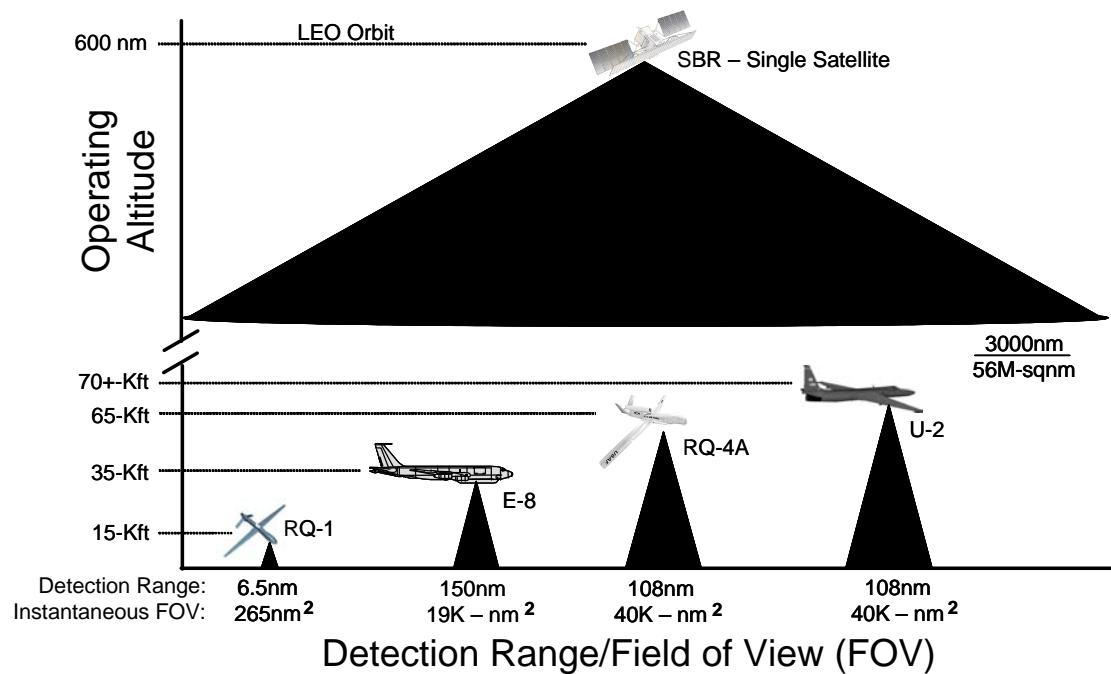


Figure 1-3: ISR Platform Coverage Comparisons. Wide Field of View (FOV) and narrow FOV assets are shown, along with their corresponding detection ranges and FOVs, which clearly differ greatly in value.

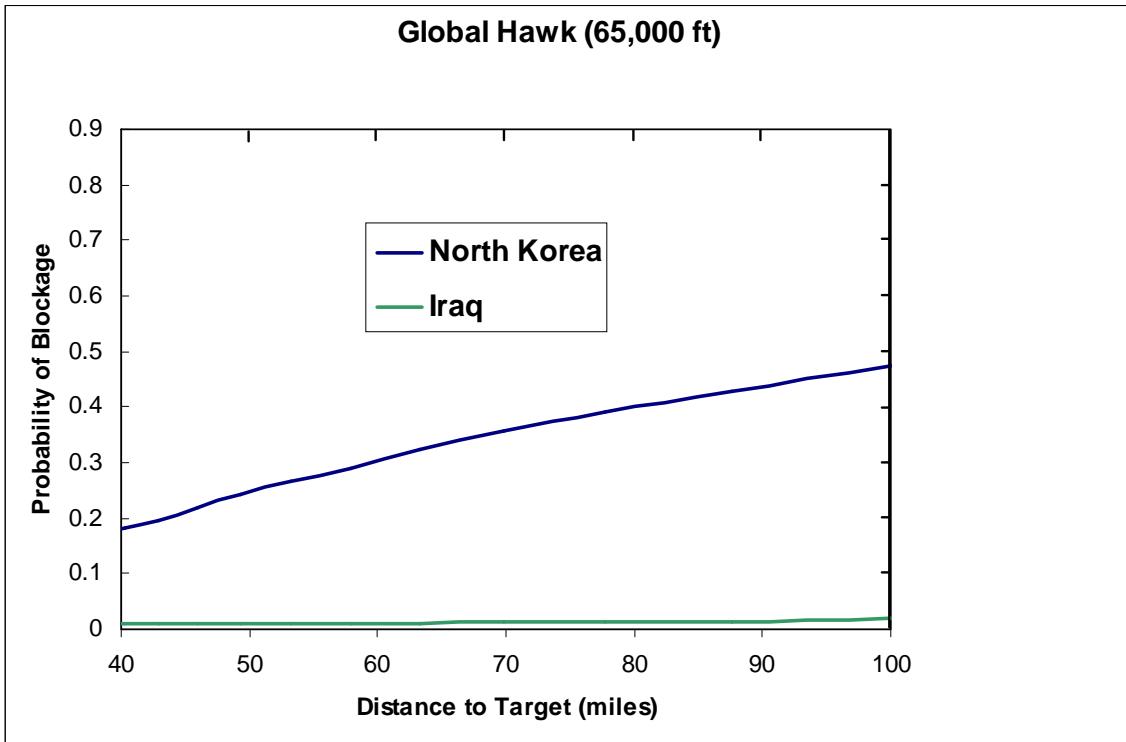


Figure 1-4: Probability of Blockage for Global Hawk UAV. Flying at 65,000 ft, the Global Hawk UAV has a much better view of the target and has less obstructions in its way. In Iraq, its probability of blockage rises only to 3% at 100 miles. In North Korea, it rises to about 48% at 100 miles from a low of 20% at 43 miles.

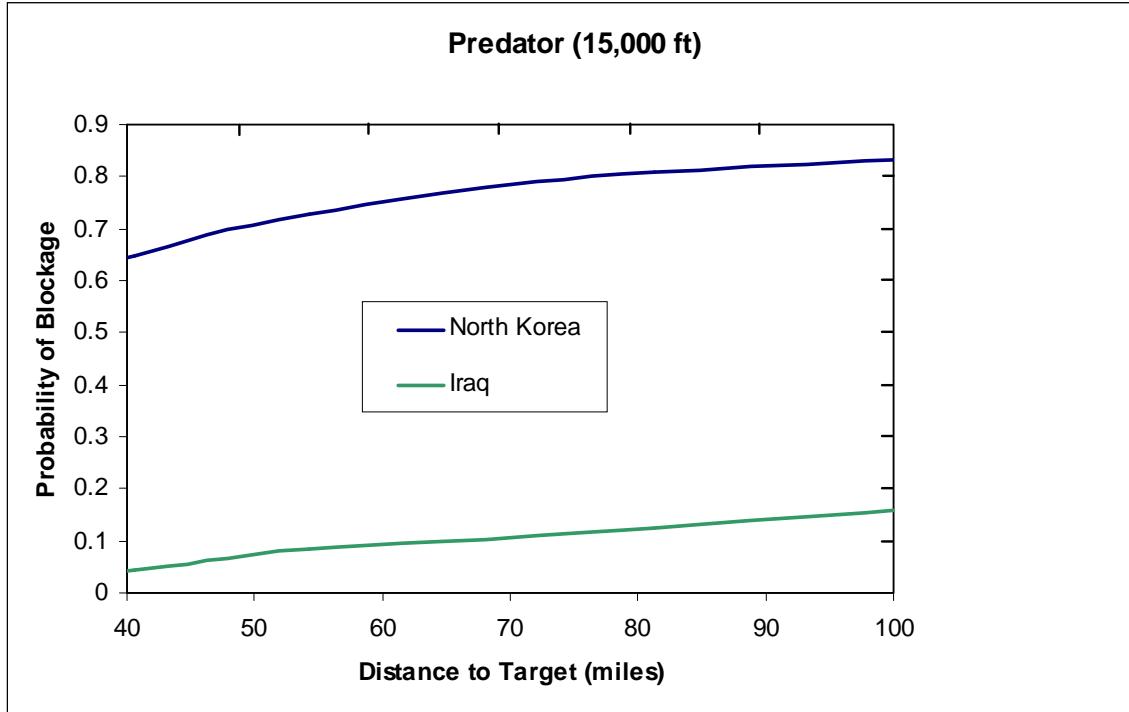


Figure 1-5: Probability of Blockage for Predator UAV. Flying at 15,000 ft, the Predator UAV in general has a more obstructed view of the target than the Global Hawk. In Iraq, its probability of blockage rises 16% at 100 miles. In North Korea, it rises to over 80% at 100 miles from a low of 65%, effectively hindering the Predator's efforts to acquire images throughout most of the country at a majority of angles.

higher. The conclusion is that the narrow FOV airborne asset must be able to travel deep inside enemy territory, and the best type of aircraft for that mission is a penetrating UAV.

1.5 Penetrating UAVs

UAVs have many unique advantages over manned systems. They are often smaller, more maneuverable, and more efficiently designed. They can operate for extended periods of time while avoiding the risk of placing human pilots in dangerous situations. UAVs have proven to be highly reliable and effective in both intelligence-gathering operations as well as missions to put bombs on target. They are discussed in more depth in Appendix B.

Once given the cue from a wide FOV sensor, the penetrating UAV must be routed to the area of interest to find the SAM site. Conversely, the SAM will do its best to shoot down the UAV. Within this problem lies a simple game with two competing sets of strategies. The UAV's strategies comprise the multitude of different paths it could take to enter the area, identify the SAM, and depart the area. The SAM's strategies involve the timing of the missile launches and the uncertainty in its location. Using the theory of games, the chances of the UAV successfully completing its mission can be analyzed.

The scope and complexity of the situation and problem described thus far is vast. Many factors contribute to the eventual UAV and SAM strategies chosen through game theoretic optimization. Figure 1-6 provides a block diagram of the essential components of the entire process. \underline{x} is defined as the set of UAV strategies, and \underline{y} is defined as the set of SAM strategies. Different performance metrics are calculated to give a resulting payoff for each strategy. Those payoffs are then used by the game theoretic optimization process to calculate the optimal UAV and SAM strategies, x and y . Each component of the block diagram in Figure 1-6 is explained in more depth in subsequent chapters.

1.6 Scenarios

The game between the UAV and SAM is analyzed in four separate cases with increasing levels of complexity. In the first case, the SAM's location is assumed to be known and fixed. This knowledge could be derived from the predictive archive, which is a collection of information

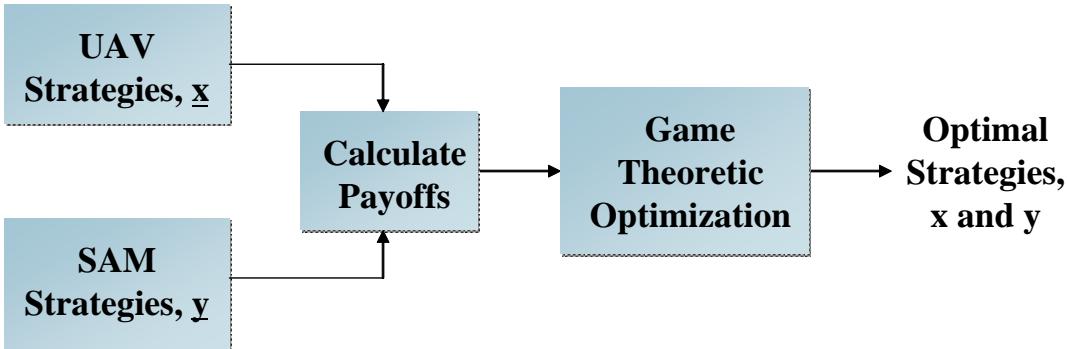


Figure 1-6: Block Diagram of Main Process. Inputs include the UAV strategies, \underline{x} , and the SAM strategies, \underline{y} . Different metrics then combine to give payoffs for each strategy. Those payoffs are then fed into the game theoretic optimization.

from past engagements and intelligence operations. The UAV's strategies will include optimal trajectories that minimize its probability of kill (probability of getting shot down) based on that known location. This case is probably the most unrealistic due to the high uncertainty in SAM locations. However, it does provide a valuable baseline for comparison to other cases.

The second case encompasses the first one but introduces uncertainty in the location of the SAM site. It is assumed that the UAV is not "smart" and thus does not take the uncertainty into account. Thus, the effect on the UAV's probability of kill and other variables is shown for increasing uncertainty in the SAM location. This case brings a certain amount of reality to the ideal scenario in case one.

The third case builds on the second case and assumes that the UAV has now increased its knowledge of the SAM and will act more intelligently. Typical battlefield intelligence from the predictive archive may suggest that the SAM is at one of N possible locations. The first scenario uses three possible SAM locations, and later analysis depicts results from five, seven, and nine possible SAM locations. Each SAM location is given a certain probability that the SAM is actually located there. Thus, the UAV must attempt to use the known probabilities and locations to optimize its trajectory more intelligently.

The fourth and final case represents the most realistic scenario. A wide FOV asset such as electrical intelligence (ELINT) satellites or Defense Support Program (DSP) satellites will pass a cue to the penetrating UAV, which must then search an area for the target. Usually

defined by an ellipse, the cued region will contain different areas with varying probabilities that the SAM is actually located there. The UAV must use its knowledge of the elliptical region and the associated probabilities to obtain an accurate target identification while minimizing its probability of kill throughout the area. These four cases embody the full spectrum of uncertainty in SAM location.

Chapter 2

UAV Strategies

The formulation of the UAV's strategies first involves defining the locations of a series of waypoints. For practical purposes, the actual space surrounding the SAM is divided into a large number of waypoints through which the UAV can travel. This simplification makes the optimization routines more tractable. The locations of the waypoints are in concentric circles from the inner radius of the SAM out to its detection radius. The waypoints are also divided vertically into a number of altitude layers. Each different combination of these waypoints results in a unique UAV trajectory. Figure 2-1 depicts this step along with the other components of this process.

After finding all possible permutations of the defined waypoints, individual paths are eliminated from the set based on limitations on UAV turning capabilities, line of sight, and terrain. These limitations are discussed later in this chapter.

2.1 Combinatorial Problems

It becomes immediately clear that a combinatorial problem arises. Consider a SAM with an inner radius of 20 km, an outer radius of 40 km, radius steps every 10 km, and height steps every 4,000 meters. That combination of parameters would yield 144 waypoints (for a five degree swath). A more realistic set of parameters might be the following: an inner radius of 20 km, an outer radius of 100 km, radius steps of 5 km, and height steps every 2,000 meters. This combination would produce 1,584 waypoints (for a seven degree swath), severely increasing the

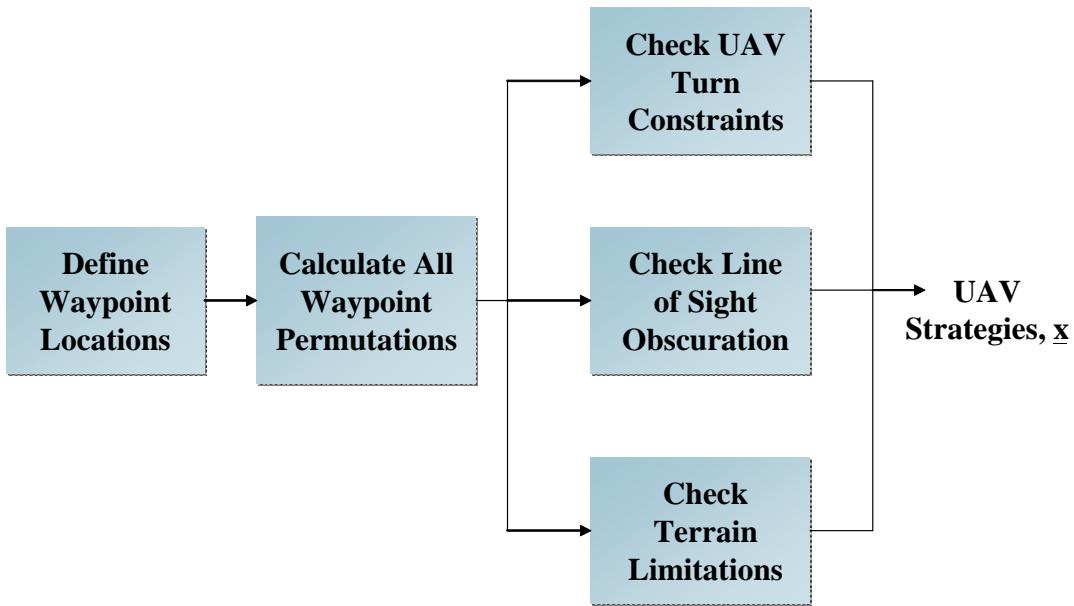


Figure 2-1: UAV Strategy Creation. Each waypoint is first defined. The set of generated waypoints are then permuted to obtain a large sample of possible UAV paths. Those paths which run into terrain, lose line-of-sight with the SAM, and surpass turning capabilities are then eliminated to give only the feasible set of UAV paths.

number of possible UAV trajectories. The number of waypoints can be calculated according to the following equation:

$$\# \text{ Waypoints} = \left[\frac{(Rad_{Outer} - Rad_{Inner})}{Step_{Rad}} + 2 \right] \cdot (Deg \text{ Swath} + 1) \cdot \left(\frac{20,000}{Step_{Height}} + 1 \right). \quad (2.1)$$

If there were only five waypoints, 120 different permutations (UAV paths) could be constructed, and that number only includes paths which pass through all five waypoints once, which is a smaller set than the total number of theoretical paths. If there were 10 waypoints, more than three and a half million permutations could be found. This set of permutations increases according to the factorial of the number of waypoints. Due to this sharp increase, the set of possible permutations for 144 or 1,584 waypoints becomes extremely large, which eliminates this method of path creation as a possibility.

2.2 Typical Aircraft Tactics

Thus, a different scheme is used to generate the UAV paths. Common sense dictates that the first strategy of any aircraft near a SAM would be to simply avoid it. In this case, the UAV's mission necessitates penetration of the SAM radius. Thus, its second strategy may be to perform terrain-following maneuvers. In other words, the UAV could hide behind hills, fly inside valleys, and use terrain to mask its presence to the SAM radar. Flying close to the ground will also help the UAV's radar echo blend in with the ground clutter, which is the radar return reflected from physical features on the ground. Other options for the UAV may be to enter the SAM radius at high altitudes, descend quickly to obtain an image, and depart at low altitude. Conversely, the UAV could enter low and depart high or perform different combinations of entering and exiting strategies [2].

One maneuver used by pilots is to "beam" the radar, which means to fly orthogonal to the radar beam [2]. This produces a zero Doppler velocity (radial velocity) and makes it appear to the radar as if the aircraft is just a part of the ground or sky. The Doppler shift describes the shift in wavelength of a moving object as opposed to what that wavelength would be if the object were at rest. This shift in wavelength can be related to the frequency of the reflected radar waves and to the velocity of the object, which is the UAV in this case. Figure 2-2

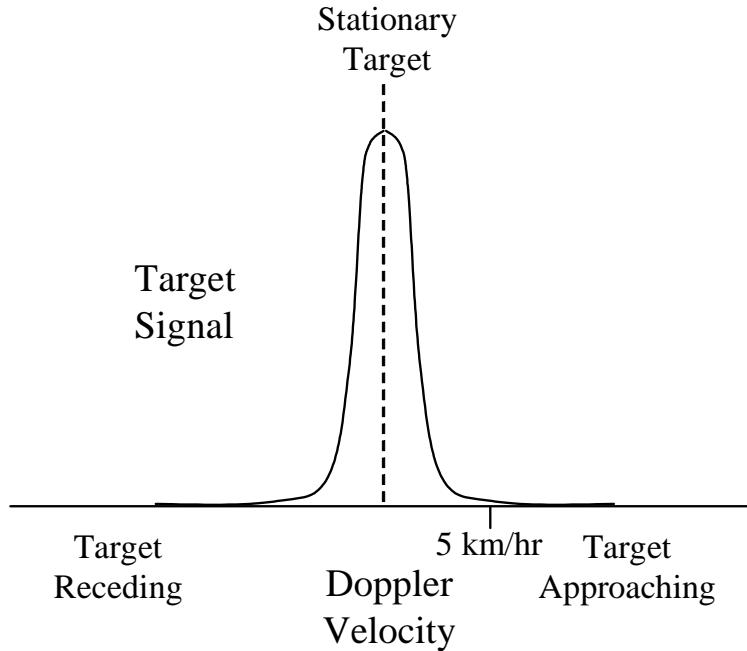


Figure 2-2: Doppler Shift. The bottom axis depicts whether the target is receding or approaching. At the frequency of stationary targets, the signal gets quite large due to the ground and other stationary objects in the path of the radar beam. To ignore those objects as clutter, the radar must establish limits on the frequency with the intent that nothing below a defined threshold is really moving. If the UAV's Doppler velocity can stay below that threshold, it can successfully avoid radar detection.

illustrates the concept of using the Doppler shift to the UAV's advantage.

The bottom axis depicts whether the target is receding or approaching. At the frequency of stationary targets, the signal gets quite large due to the ground and other stationary objects in the path of the radar beam. To ignore those objects as clutter, the radar must establish limits on the frequency with the intent that nothing below a defined threshold is really moving. If the UAV flies orthogonal to the radar beam, its radar echo is enveloped by the signals of stationary targets. This makes it appear to the radar as if the UAV is just a part of the ground. However, the UAV's Doppler velocity must stay below the radar-defined threshold, which represents the lowest possible speed at which a target can be detected. A value of 5 km/hour is shown in Figure 2-2. The concept of the Doppler shift and a full discussion of radar is included in Appendix C.

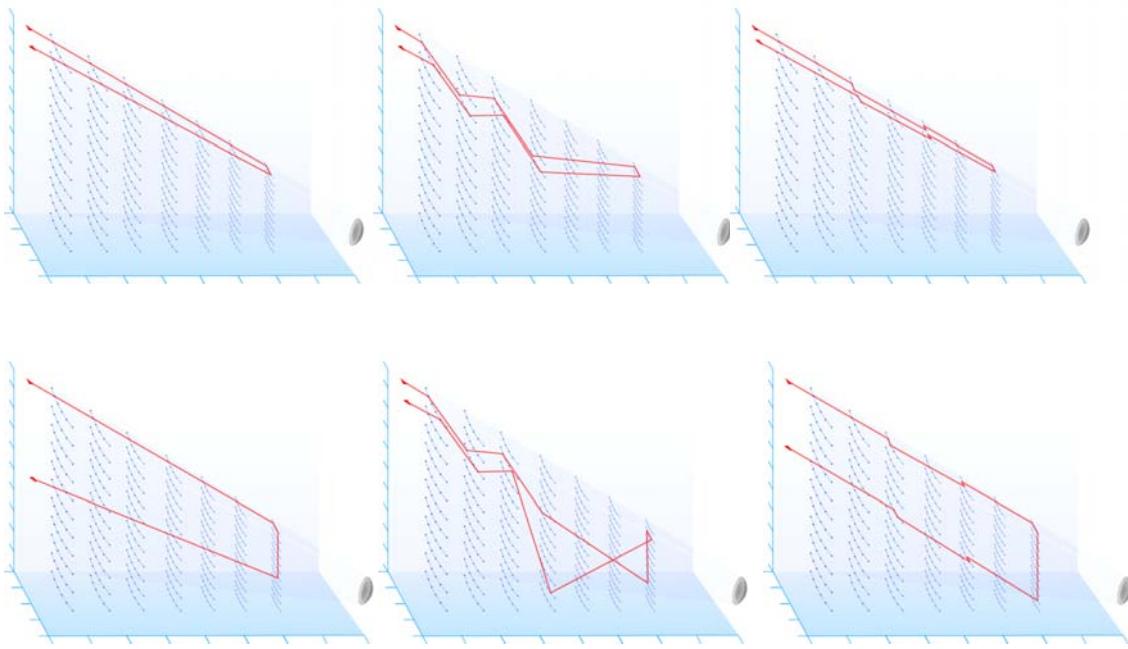


Figure 2-3: Possible Types of UAV Paths. The UAV could come in high and depart low, come in low and depart high, or come in and depart at the same altitude. Additionally, it could travel deep into the SAM radius or stay on the edge. It could fly straight in, use a zigzag pattern, or use the beaming maneuver frequently. Regardless, the beaming maneuver is always used for every path while taking the image.

2.3 UAV Strategy Formulation

Using many of the tactics outlined in the previous section, the UAV paths can then be generated. The resulting types of trajectories (Figure 2-3) closely resemble the previously discussed pilot maneuvers. The UAV could enter the SAM radius at a certain height and depart at the same height. Its path could be a straight line in and out to give the fastest mission completion time. The path could be zigzagged to present difficulties for the SAM radar, or it could use the beaming technique both while approaching the SAM and while departing. In addition, the UAV can start a trajectory at high altitudes and return at low altitudes or start at low altitudes while returning at high altitudes. All of these various UAV paths are depicted in Figure 2-3. The characteristic that remains constant throughout all the paths is the fact that while taking the image, the UAV is beaming the radar. Since the majority of sensors are side-looking, this assumption also simplifies the problem of having the SAM in the appropriate field of view to

take the image.

2.3.1 Turn Constraints

The set of created paths is then sorted to eliminate the paths that are not possible due to various reasons. Those paths which are not possible due to the turn constraints of the UAV are first eliminated. The turn limitation is mostly dependent on the speed of the UAV, and this constraint usually eliminates about 25% of the paths. The first step is to calculate the vector between two sets of waypoints as follows:

$$\begin{aligned} q_1 &= w_{i+2} - w_{i+1} \\ q_2 &= w_{i+1} - w_i, \end{aligned} \tag{2.2}$$

where w_i is a waypoint location, and q_i is a vector. The angle between the two resulting vectors is then found using the formula for the dot product:

$$\theta = \arccos \left(\frac{q_1 \cdot q_2}{|q_1| |q_2|} \right), \tag{2.3}$$

where θ is the angle between the vectors q_1 and q_2 . The minimum possible turn radius between these points is then calculated as follows:

$$r_{\min} = \frac{v^2}{g \sqrt{n_{\max}^2 - 1}}, \tag{2.4}$$

where v is the velocity of the aircraft, g is the acceleration of earth's gravity (9.8 m/s^2), and n_{\max} is the maximum sustainable load factor. If a human is in the cockpit, n_{\max} would be nine times the value of g . Once r_{\min} is found, it can be inserted into Equation 2.5:

$$\theta_{\max} = \arctan \left(\frac{v^2}{g \cdot r_{\min}} \right). \tag{2.5}$$

The resulting value of θ_{\max} is then compared with the previously calculated value of θ from Equation 2.3. If θ is greater than θ_{\max} , the path containing that specific set of waypoints is then eliminated. This process is repeated for every set of waypoints along every possible UAV

path.

2.3.2 Line of Sight Obscuration

The next constraint deals with visibility. To obtain an image, the UAV must have the SAM in its line of sight (LOS). In other words, no terrain features can obstruct the straight line connecting the UAV's sensor to the SAM location. Thus, the next step involves using Digital Terrain Elevation Data (DTED) sets to ensure clear LOS is present, which eliminates on average a third of the remaining paths. DTED sets were developed by the National Imagery and Mapping Agency (NIMA), now the National Geospatial-Intelligence Agency (NGA), in support of military applications. They provide a matrix of terrain elevation values for most parts of the world in three different levels. Level 0 DTED provides elevation data at a spacing of roughly one kilometer. Level 1 DTED (used by this author) provides data every 100 meters, which is similar to a 1:250,000 scale map. Finally, Level 2 DTED provides data roughly every 30 meters.

The imaginary lines connecting the UAV and the SAM is found for the set of points during which the UAV will be taking an image of the SAM. The height of the terrain is found at each point and compared with the UAV altitude. This process is then repeated incrementally with a small step size along the imaginary line until the SAM is reached. If the height of the terrain interferes just once with the imaginary line, then the associated UAV path is eliminated from the set.

2.3.3 Terrain Limitations

The final step in the process of creating the UAV strategies is to ensure that the UAV does not fly into any terrain. Using the DTED sets once again, this step compares the UAV's height, h , at each point along each UAV path with the terrain elevation at that point, h_{\min} . If h is less than h_{\min} , the associated UAV path is eliminated from the set. Interpolation between waypoints is performed to ensure accuracy. This step eliminates another tenth of the possible paths. These numbers are simply rough averages since the actual number of path eliminations depends entirely on the UAV and the region's terrain. However, it is quite feasible to have an initial set of 330 paths, for example, be cut down to only 146 valid ones.

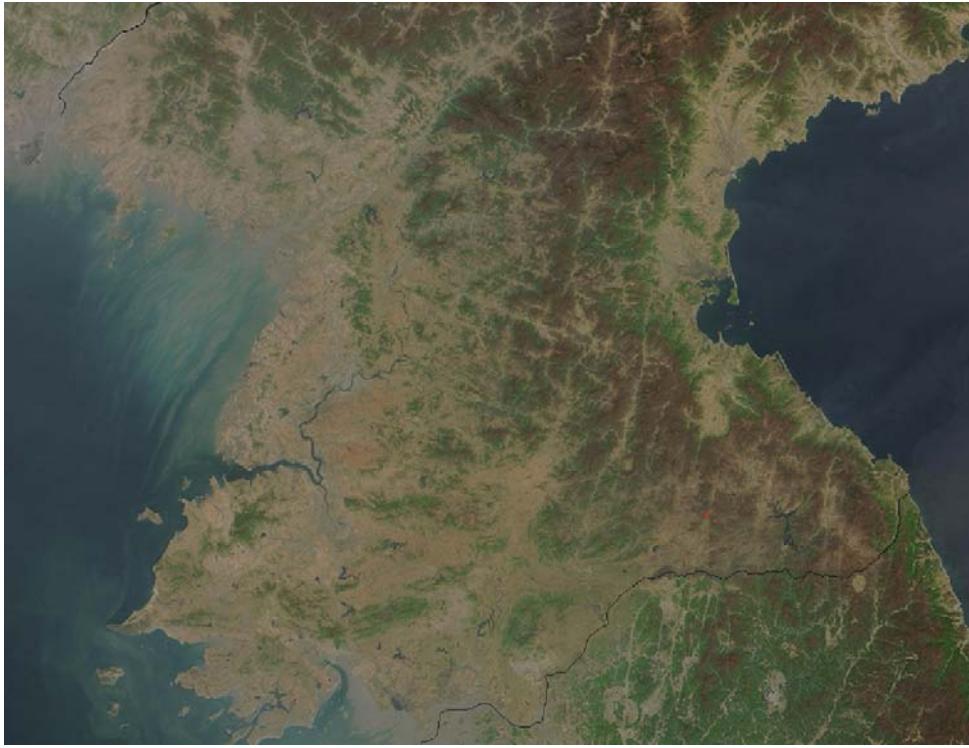


Figure 2-4: Terrain Map of North Korea (NASA). The intensity and variation of light brown, dark brown, and green spots gives a view of the roughness of terrain in North Korea. This environment is a good challenge to test out a program using terrain-avoidance code.

For this simulation, the SAM’s location (initially fixed) was chosen to be at 126.75 deg E longitude and 38.75 deg N latitude, which lies directly in the center of North Korea. This location was chosen mostly because of the rough variety of terrain encountered, which provides good challenges to terrain avoidance and line of sight calculations. Figure 2-4 provides an excellent view of the mountainous regions in North Korea.

2.4 Image Quality

The UAV’s mission is to identify a target after receiving a cue from a wide FOV asset. Therefore, an important metric to describe each UAV path is the quality of the image taken. For many years, image scale, resolution, and other image quality measures were the only tools available to evaluate the quality of an image. However, they were never adequate enough to predict

the interpretability of an image. In the early 1970s, a collaboration of government and contract workers developed the National Imagery Interpretability Rating Scale (NIIRS) for the Imagery Resolution And Reporting Standards (IRARS) Committee [23]. Image analysts first defined a standard set of interpretation tasks that they were commonly asked to perform. They were then given sets of images with pre-determined and varying quality and asked to describe the interpretation tasks that could be performed on the images. The original 10-level scale grew out of this study. Over the years, it has evolved many times, especially since many objects in the original NIIRS list became outdated [23]. NIIRS is now the standard scale used by image analysts, scientists, designers, and managers. It is task-based in that each NIIRS level implies different tasks that can be performed by an analyst [23]. With this tool, one can asses image quality as well as provide quantifiable means for expressing sensor system requirements. Four separate NIIRS scales have been developed: Radar NIIRS, Visible NIIRS, Infrared (IR) NIIRS, and Multispectral (MS) NIIRS [23]. These scales consist of 10 graduated levels (0 to 9) with every increase in level signifying a similar increase in image quality and in the difficulty of the interpretation task required by the analyst. More information on the history and development of NIIRS can be found in Appendix D.2.1, and lengthy descriptions of the four scales are located in Appendix E. Additionally, information on modern sensors such as synthetic aperture radar (SAR) or electro-optical (EO) sensors is found in Appendix D.

The widespread use of NIIRS by sensor system developers led logically to the necessity of a method or technique that will accurately predict a NIIRS level based on sensor attributes before the actual sensor is built. Developed in the 1980s but not formally released until 1994, the General Image Quality Equation (GIQE) predicts NIIRS as a function of predicted image scale, sharpness or resolution, and signal-to-noise (S/N) ratio [22]. The GIQE was initially developed under the IRARS Committee and used a regression modeling approach. Ten image analysts conferred NIIRS ratings on samples of EO imagery. The characteristics of those samples were then used to develop a regression model that predicted NIIRS as a function of perceptual quality attributes of scale, resolution and sharpness, contrast, and noise [22]. The GIQE was released to the UAV community in 1994 but has undergone many changes and improvements since that time.

The original EO GIQE is defined in Equation 2.6.

$$NIIRS = 11.81 + 3.32 \log_{10} \left(\frac{RER_{GM}}{GSD_{GM}} \right) - (1.48H_{GM}) - \left(\frac{G}{S/N} \right). \quad (2.6)$$

RER_{GM} is the geometric mean of the Relative Edge Response (RER), which relates to perceived sharpness or acutance of the image. GSD_{GM} is the geometric mean of the Ground Sampled Distance (GSD), which measures both scale and resolution. EO system post-processing techniques involve modulation transfer function compensation (MTFC), which increases edge response and noise [22]. Thus, other terms were added to the GIQE to account for these increases. H_{GM} is the geometric mean of the height of the overshoot due to edge sharpening. G is the noise gain due to edge sharpening, and the S/N provides for some modeling of contrast. The GIQE does not account for bandwidth compression (BWC), softcopy image output prediction, or inclement weather [22].

RER is the slope of the edge system response and is found by measuring two points that are 0.5 pixels from the edge using a normalization over the range of 0 to 1. GSD is found with the following equation 2.7:

$$GSD = \frac{\left(\left(\frac{\text{pixel pitch}}{\text{focal length}} \right) \times \text{slant range} \right)}{\cos(\text{look angle})}. \quad (2.7)$$

Both GSD and RER are computed along the X and Y axes, and then the geometric means are found. H is found by measuring the maximum value over the range 1.0 to 3.0 pixels from the edge at 0.25 pixel increments. The mean of this term is also found using the values along the X and Y axes.

Substituting physical parameters wherever possible and using simplifications from Equation 2.7, Equation 2.6 then becomes the following:

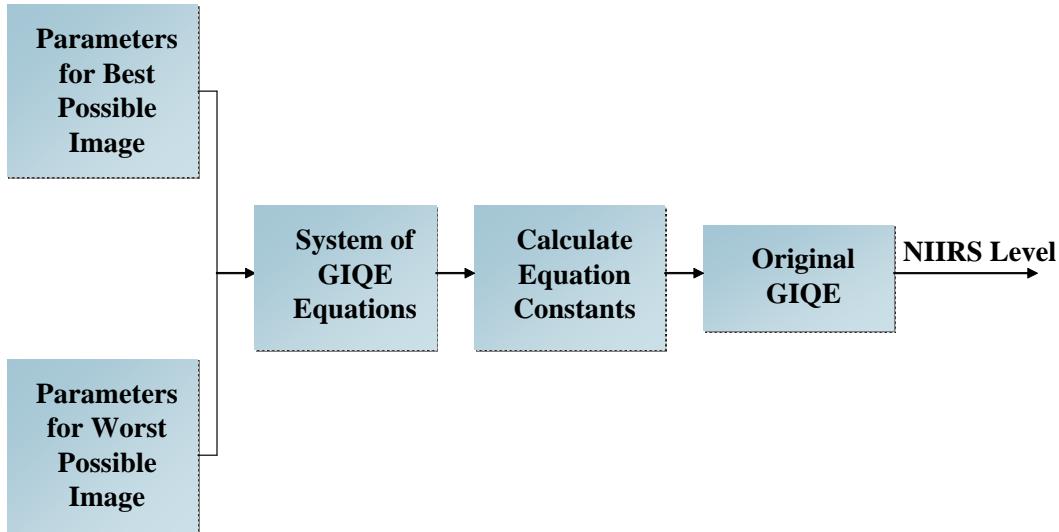


Figure 2-5: Block Diagram of Image Quality Calculations. The parameters for the best and worst image quality are input into the GIQE to form a system of equations. The equation constants are then solved for and inputted into the GIQE, which outputs the NIIRS Level as the result of this process.

$$\begin{aligned}
 NIIRS = & 11.81 + 3.32 \log_{10} \left(\frac{\min(\text{abs}(Az - 90), \text{abs}(Az - 270))}{R} \right) \\
 & + 2 \cos(\text{abs}(Dep - 45)) - 1 + C_1 \\
 & - C_2 \cdot \left(\frac{1}{\log_{10} \left(2.0736 \times 10^{30} \cdot \left(\frac{10^{10}}{R^4} \right) \right)} \right),
 \end{aligned} \tag{2.8}$$

where C_1 and C_2 are equation constants. The main physical parameters that contribute to the quality of an image are the range, R , the depression angle, Dep , and the azimuth angle, Az . These variables occur repeatedly in Equation 2.8.

Using the previously defined relationships, the process of determining the image quality, I_q , for each UAV trajectory is shown in Figure 2-5. It will be assumed here that I_q is synonymous with NIIRS Level. The best (highest image quality) and worst (lowest image quality) values of the three main physical parameters are outlined in Table 2.1.

The "worst" parameter values are substituted into Equation 2.8 to produce the top line

Parameters	Best	Worst
Range (km)	20	200
Azimuth (deg)	90	180
Depression (deg)	45	0

Table 2.1: Best and Worst Parameter Values

of Equation 2.9. The "best" parameter values yield the bottom line of Equation 2.9. The following system of GIQE equations can help derive a baseline for I_q calculations.

$$\begin{aligned} 0 &= -5.87 + C_1 - (C_2 \cdot 0.0523) \\ 6.5 &= -1.47 + C_1 - (C_2 \cdot 0.0433). \end{aligned} \quad (2.9)$$

The values of zero and 6.5 are NIIRS values that correspond to the worst and best parameter values, respectively. The value of 6.5 value was obtained using published Raytheon data about the capabilities of the Global Hawk's Integrated Sensor Suite [16]. Solving the system of equations yields $C_1 = 17.98$ and $C_2 = 231.31$. Substituting these values into Equation 2.8 gives the final modified equation to calculate image quality.

$$\begin{aligned} NIIRS &= 11.81 + 3.32 \log_{10} \left(\frac{\min(\text{abs}(Az - 90), \text{abs}(Az - 270))}{R} \right) \\ &\quad + 2 \cos(\text{abs}(Dep - 45)) - 1 + 17.98 \\ &\quad - 231.31 \cdot \left(\frac{1}{\log_{10} \left(2.0736 \times 10^{30} \cdot \left(\frac{10^{10}}{R^4} \right) \right)} \right). \end{aligned} \quad (2.10)$$

Using a range of values for the two most influential parameters, azimuth and range, a color map (Figure 2-6) was created to show the variation in image quality. Clearly, the best images are obtained when the aircraft is at 90 or 270 degrees and as close to the SAM as possible. Using the azimuth, range, and depression of the UAV at the points where an image is being taken, an I_q is thus calculated for every UAV strategy.

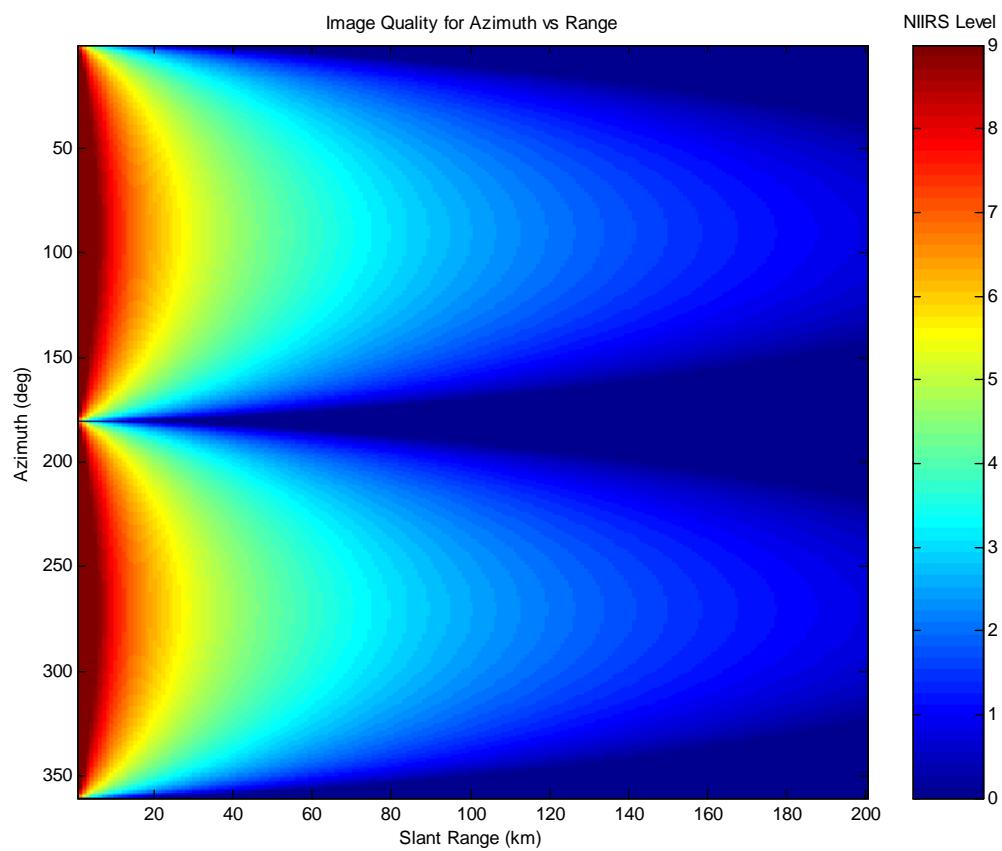


Figure 2-6: Color Map of Image Quality with Varying Azimuth and Depression. A range of values for azimuth and range, the most influential parameters, are used to show the variation in image quality. Clearly, the best images are obtained when the aircraft is at 90 or 270 degrees (perpendicular to the SAM) and as close to the SAM as possible.

Chapter 3

SAM Strategies

In a similar fashion, the SAM also creates its own strategies to counter those of the UAV. One potential strategy is to fire a missile at the UAV when it first detects it. Another strategy is more deceptive. The SAM will wait until the UAV has acquired its image and is outbound before firing a missile. This second strategy assumes the UAV will be more vulnerable after completing over half of its trajectory without being fired upon. Both of these SAM strategies involve the processes of detecting the UAV and tracking it, among other components.

3.1 Detections

Detecting the UAV is not a simple task. A block diagram of this process is located in Figure 3-1. The first step is to find a UAV (target) model. The model used here was created by the use of the Lincoln Lab Component Object Radar Signatures (LL CORS) toolbox. LL CORS is component-based, object-oriented Matlab code that coherently combines radar scattering from various components on an aircraft (or any target) to produce a model of the UAV's Radar Cross Section (RCS).

The RCS, σ , of a target is the magnitude of the echo signal reflected by the target and received by the radar [31]. It has also been defined more formally as "the area intercepting that amount of power which, if radiated isotropically, produces the same received power in the radar" [30]. RCS is an area, and its units are typically in meters squared or decibels. The UAV's strategies are created with the intent to minimize its RCS as much as possible. The

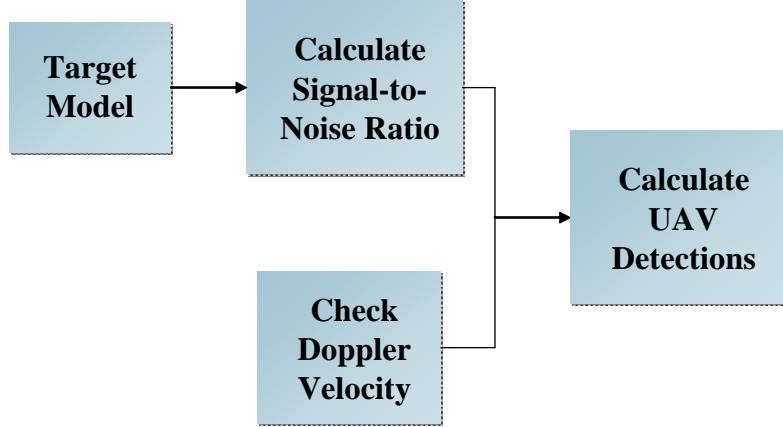


Figure 3-1: Block Diagram of Detection Process. The model of the target is used to calculate the S/N. Comparing this value to the minimum S/N gives a matrix of detection values. The Doppler velocity minimum is also checked to determine whether the UAV's Doppler velocity is less, and thus the UAV would avoid detection.

SAM requires a certain level of RCS to even be able to detect the target. A more lengthy discussion of RCS is located in Appendix C.2.

The LL CORS code involves the use of surface scattering, diffracting edges, point scattering, and special ring and joint scattering. Ready-made target models can be processed through the program, or geometric shapes can be generated and combined within LL CORS to model a certain target. Using a combination of cylinders and flat panels, a model UAV was created for this simulation using LL CORS. That model is shown in Figure 3-2. This model was intended to be similar in size and shape to the Predator UAV, although dimensions and proportions are not exact. The model was then input into the RCS scattering simulations of LL CORS to obtain a model of the UAV's RCS from different angles. One of the outputs was the UAV's RCS as aspect angle is varied. An aspect angle of zero degrees means that one is looking head-on at the UAV. An aspect angle of 90 degrees signifies looking directly at the side of the aircraft in the same plane as the front wings, and an aspect angle of 180 degrees means a tail view. This graph is shown in Figure 3-3.

The model RCS remains relatively constant throughout the entire front section. 90 degrees shows the sharpest increase in RCS to over 35 dB due to the sharp edges of the wings. The RCS then slopes off significantly as the tail comes into view with lows of -35 dB. Only one

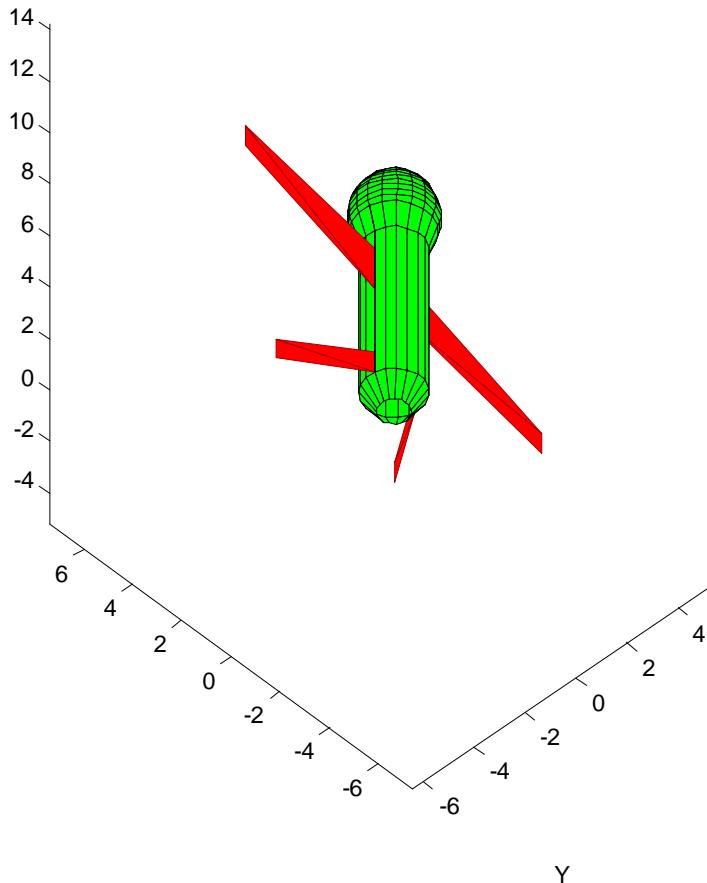


Figure 3-2: UAV Model. Using a combination of cylinders and flat panels, this model UAV was created for the simulation using LL CORS. This model was intended to be similar in size and shape to the Predator UAV, although dimensions and proportions are not exact. It includes a rounded, bulbous nose, a cylindrical body, two long-spanned front wings, and two smaller back wings with increased dihedral.

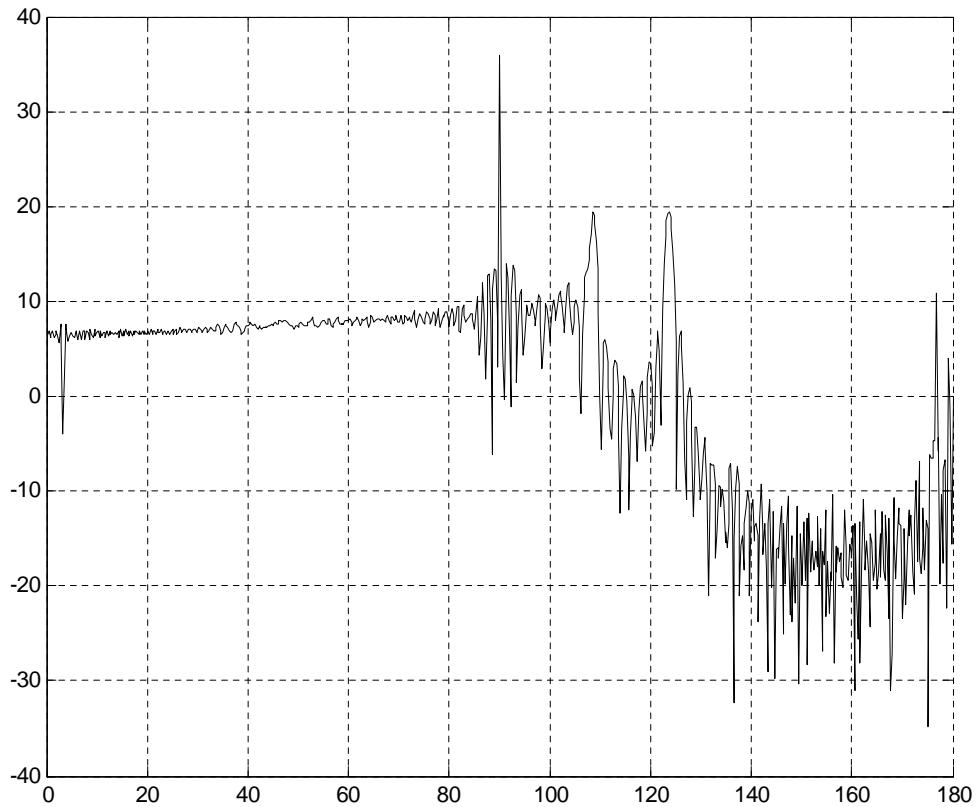


Figure 3-3: RCS of UAV Model with Varying Aspect Angle. The RCS remains relatively constant throughout the entire front section. 90 degrees shows the sharpest increase in RCS to over 35 dB due to the sharp edges of the wings. The RCS then slopes off significantly as the tail comes into view with lows of -35 dB. Only one side of the aircraft is shown because the model is completely symmetrical. Thus, a graph of RCS from 180 to 360 degrees would mirror this graph, except in reverse. One other item to note is that the scattering was analyzed at a frequency of ten GHz. This value is roughly similar to common X-band frequencies, which are prevalent in fire-control radar systems.

side of the aircraft is shown because the model is completely symmetrical. Thus, a graph of RCS from 180 to 360 degrees would mirror the graph in Figure 3-3, except in reverse. One other item to note is that the scattering was performed at a frequency of 10 GHz. This value is roughly similar to common X-band frequencies, which is prevalent for SAM systems using fire-control radar. However, different radars use different frequencies. Figure 3-4 shows the RCS variation of the UAV model at an aspect angle of 90 degrees for frequencies in the range of 5 to 7 GHz. The RCS is shown to vary by about 2 dB. Although this variation does not seem significant, different values of aspect angle may yield larger deviations for different frequencies. In Figure 3-5 RCS is plotted on a color coded map versus azimuth and depression angle. Clearly, the UAV's RCS averages around 10 dB except for lows at the tail end and short peaks at each wing. Those peaks bring about especially high RCS values when the depression angle is at 90 or 270 degrees, which is looking directly at the top and bottom of the aircraft, respectively. This view presents the largest area to the radar. Depression angles of zero or 180 degrees signify looking directly at the sides of the aircraft.

This model can then give an excellent approximation for the RCS of the UAV at any point along its paths. Following along with the diagram in Figure 3-1, S/N is then calculated according to the radar range equation, shown here in Equation 3.1.

$$S/N = \frac{P_t G^2 \lambda^2 \sigma}{(4\pi)^3 R^4 k T_s B_n L}. \quad (3.1)$$

The radar equation relates the performance of a radar to its design parameters. Specifically, it connects the properties of the target, the characteristics of the radar, and the attributes of the propagation medium to obtain the range of the target and other values. It is mainly used to determine the maximum range at which a target can be detected and is often referred to as the radar range equation. It can also be modified to yield the radar equation for track in Equation C.15. S/N is essentially the ratio of the power received by the SAM's radar to the power of the noise (unwanted returns from other objects), and it is the standard measure of a radar's ability to detect a target at a given range. Unfortunately, specific values for power, P_t , gain, G , and other radar characteristics are held tight. Thus, an educated assumption had to be made that with a 10 dB S/N, a target with a zero dB RCS could just be detected at 120

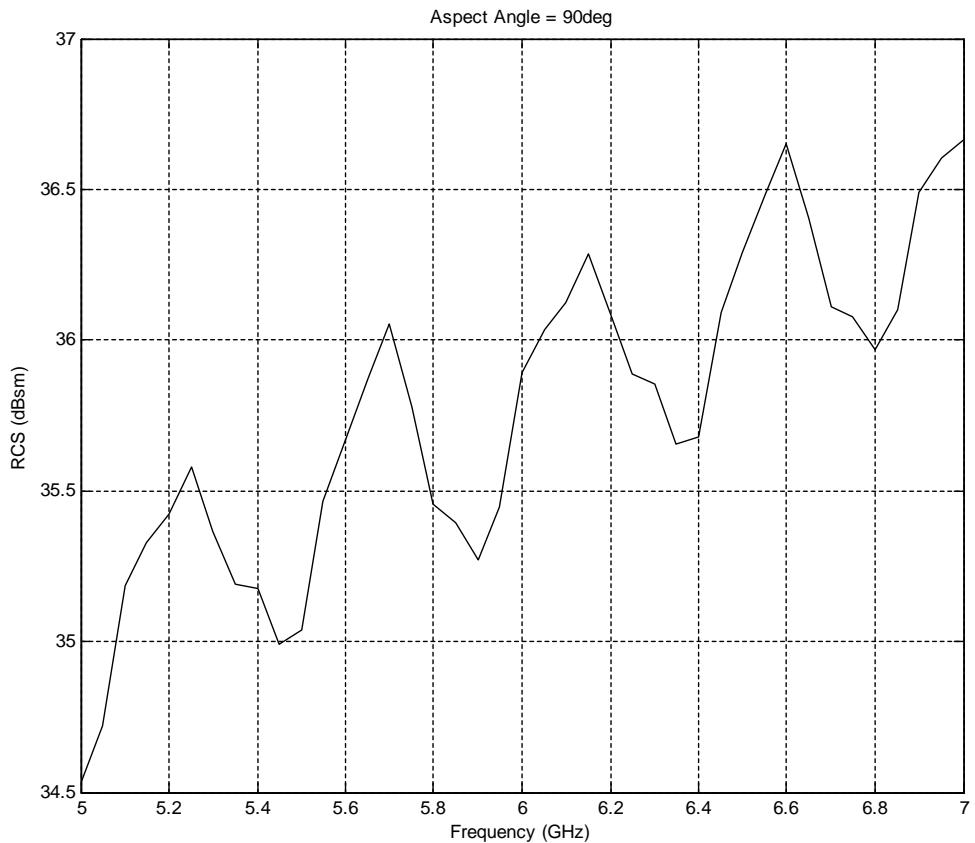


Figure 3-4: RCS of UAV Model as Frequency Varies. The frequencies at which radar waves propagate vary somewhat from radar to radar. This figure shows the RCS variation of the UAV model at an aspect angle of 90 degrees for frequencies in the range of five to seven GHz. The RCS is shown to vary by about two dB.

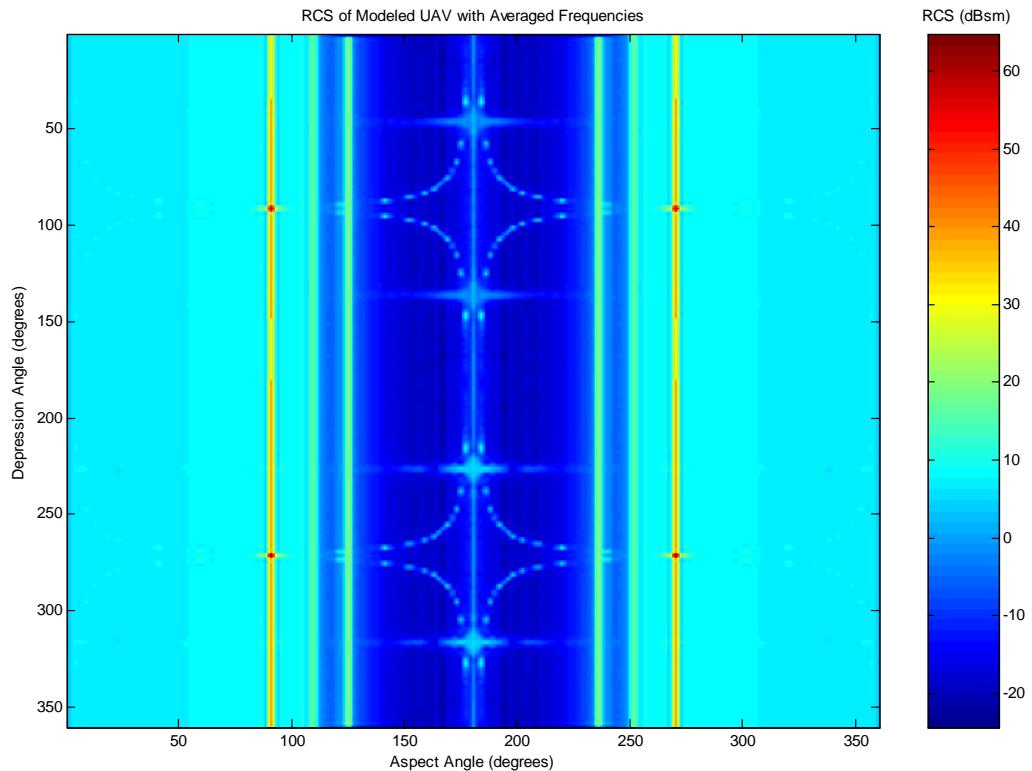


Figure 3-5: Color Map of Model RCS with Varying Depression and Azimuth. Clearly, the UAV's RCS averages around 10 dB except for lows at the tail end and short peaks at each wing. Those peaks bring about especially high RCS values when the depression angle is at 90 or 270 degrees, which is looking directly at the top and bottom of the aircraft, respectively. This view presents the largest area to the radar. Depression angles of zero or 180 degrees signify looking directly at the sides of the aircraft.

km. This results in the simplification of Equation 3.1 to the following:

$$S/N = \left[\frac{P_t G^2 \lambda^2}{(4\pi)^3 k T_s B_n L} \right] \frac{\sigma}{R^4} = \frac{C_{S/N} \cdot \sigma}{R^4}, \quad (3.2)$$

where $C_{S/N}$ is equal to 2.0736×10^{30} . In this way, using the UAV's RCS, σ , and range, R , from the SAM at every point, the S/N can be found as well. This equation can also be written in the following manner:

$$\frac{S/N}{(S/N)_o} = \frac{C_{S/N} \cdot \left(\frac{\sigma}{\sigma_o}\right)}{\left(\frac{R}{R_o}\right)^4}, \quad (3.3)$$

where $(S/N)_o = 10 \text{ dB}$, $\sigma_o = 0 \text{ dB}$, and $R_o = 120 \text{ km}$. To provide an illustration of this part of the detection process, a sample path was chosen (Figure 3-6). On this path, the UAV travels (while slowly descending) at an altitude of about 5,000 m using the beaming technique periodically throughout its trajectory in and out of the SAM radius. The S/N of this path was then graphed versus time to give Figure 3-7, which clearly shows the RCS peaks as the UAV turns to face the SAM head on after recovering from periods of beaming.

The detection cutoff value of 10 dB essentially describes the lowest detectable S/N for detecting a target. A majority of points along the path in Figure 3-7 would be treated as detections because they are above that 10 dB threshold. Those points below 10 dB would be ignored as part of the ground or atmospheric clutter and treated as non-detections.

A final check is made to see if the UAV's Doppler velocity is greater or less than the Doppler cutoff at every point. The Doppler cutoff was previously discussed and represents the lowest speed at which a target can be detected. The Doppler velocity is found using the following equation:

$$v_D = v_i \cdot \cos(\phi_i), \quad (3.4)$$

where v_i is the UAV velocity and ϕ_i is the angle between the UAV's heading and the line connecting the UAV and the SAM. If v_D is greater than 5 km/hr, the Doppler cutoff, then the detection remains unchanged. If less, a detection is automatically changed to a non-detection because the object is assumed to be stationary by the radar.

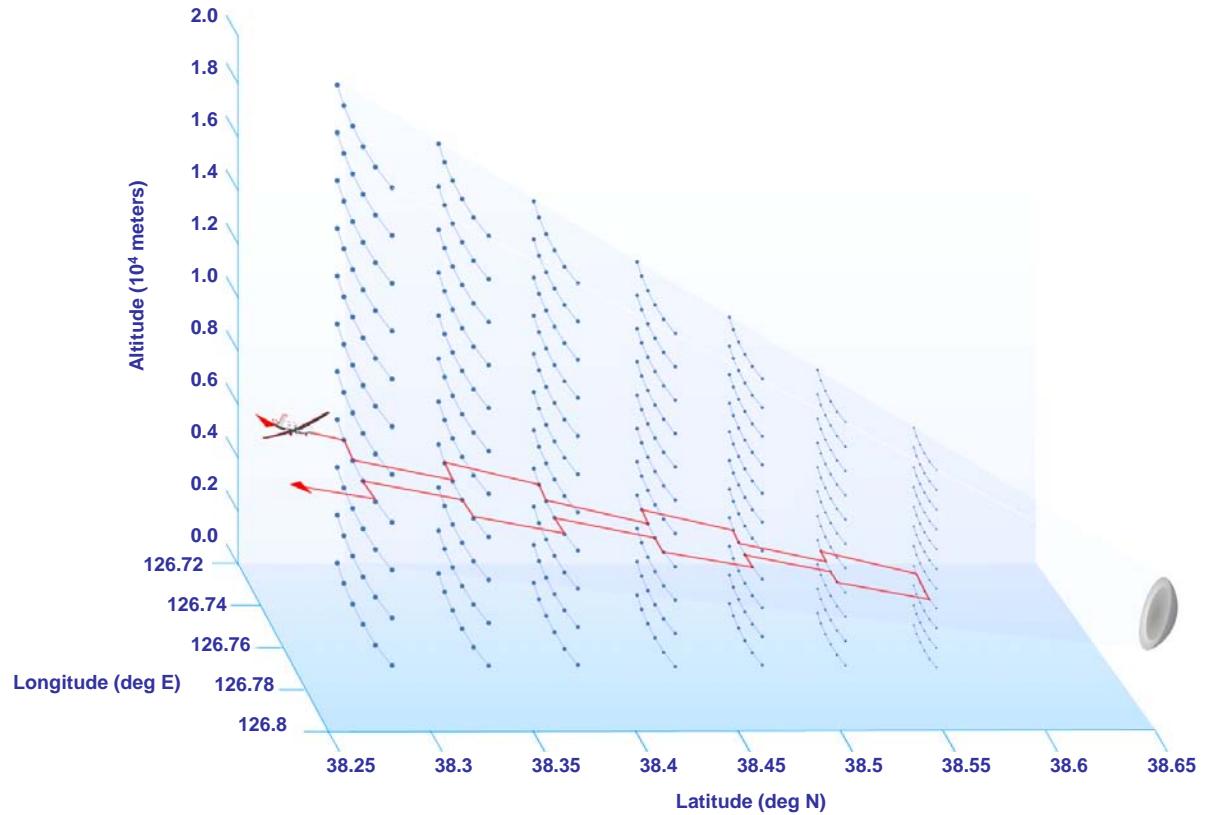


Figure 3-6: Sample UAV Trajectory. This path slowly descends on one level of a SAM's hypothetical sphere encompassing its detection range. The UAV's altitude averages around 5,000 meters, and it uses the beaming technique periodically throughout its trajectory in and out of the SAM radius.

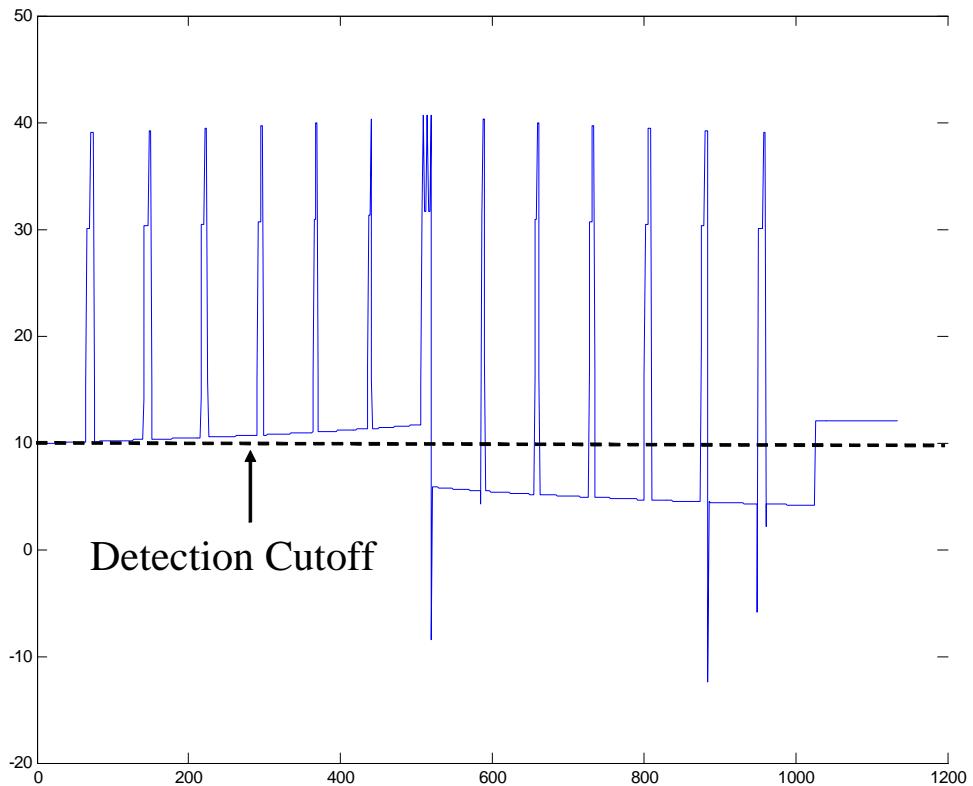


Figure 3-7: S/N of Sample Trajectory. This graph plots S/N on the left axis and elapsed time on the bottom axis. When the UAV beams the SAM, its S/N stays around 10 dB, but when it turns back toward the SAM, the S/N shoots up to about 40 dB. This trend continues until the UAV leaves the SAM detection radius and shows how easily detectable the UAV would be on this path. If the detection cutoff were 10 dB, then a majority of points along this path would be treated as detections because they are above the defined threshold. Those points below 10 dB would be ignored as part of the clutter and treated as non-detections.

3.2 Probability of Kill

Being detected, however, does not necessarily signify that the UAV will be shot down. Another process must turn detections into probabilities of kill (P_k), or the probability that the UAV will be shot down. This is where the SAM strategies are broken down further. The first strategy used is for the SAM to shoot a missile at the incoming UAV upon first detection. The time of that first detection occurs at the minimum detected value in the specific path's detection matrix (D), $t_{start} = t(D_{min})$. The distance between the SAM and UAV at that detection, $d_i = d(D_{min})$, is then found. Using the following equation, the time until missile impact is found:

$$t_I = \frac{d_I}{v_M}, \quad (3.5)$$

where v_M is the velocity of the missile. Using Equation 3.6, the P_k of that particular UAV/path combination is then calculated to be proportional to the amount of time the UAV is detected throughout the flight time of the missile until impact:

$$P_k = \frac{\# \text{ Detections}}{t_I - t_{start}} = \frac{\# \text{ Detections}}{t_{end}} = \frac{\sum_{i=t_{start}}^{t_I} D_i}{t_{end}}. \quad (3.6)$$

Also, the "first detection" must not be a momentary detection or some rare occurrence. The SAM must have established a track on the UAV, which is determined here to be 10 or more detections in a row.

The second SAM strategy is more deceiving. It attempts to lure the UAV inside its radius by not firing a missile until the UAV is determined to be outbound. This is found by calculating the Doppler velocity, as in Equation 3.4, which is negative if the UAV is getting further away. t_{start} is then set equal to the time at which the first negative Doppler velocity is calculated, $t(-Doppler)$, and t_I is calculated similarly to Equation 3.5:

$$t_I = \frac{d_{-Doppler}}{v_M}. \quad (3.7)$$

In this second case, the P_k is the amount of time the UAV is in track divided by the total flight time of the missile, as in Equation 3.6. A block diagram of this process is shown in Figure 3-8, with strategy one on the top and strategy two on the bottom.

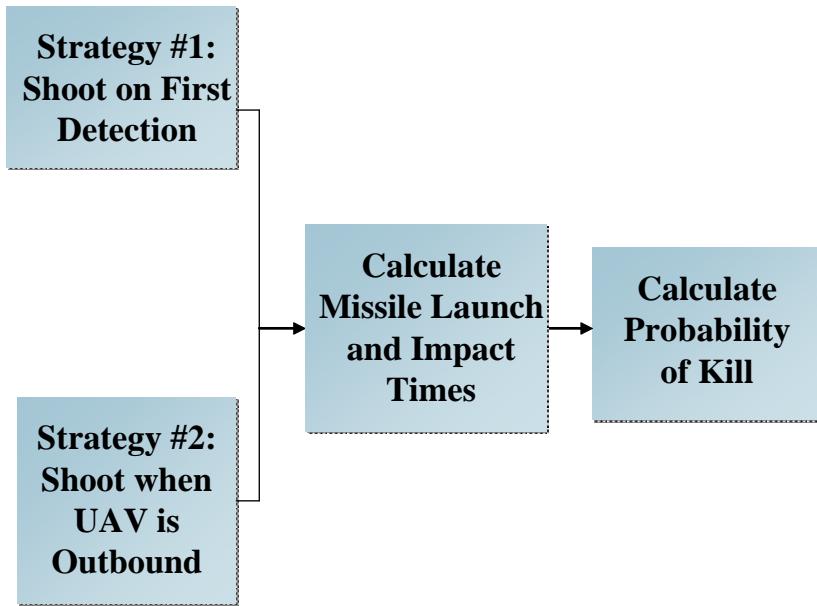


Figure 3-8: Block Diagram of Kill Probability Formulation. Two SAM strategies are utilized. The strategy at the top left is where the SAM shoots at the UAV upon first detection. The strategy at the bottom left is where the SAM shoots at the UAV when it is outbound. This is a more deceiving strategy and necessitates the calculation of the Doppler frequency. The amount of time that the UAV is in track from missile launch until missile impact for both strategies is proportional to the probability of kill that is the end output of this process.

Regarding the sample trajectory discussed previously, the UAV's P_k was 0.84 for strategy one and about 0.79 for strategy two. Thus, a P_k is calculated for every UAV strategy versus each SAM strategy.

Chapter 4

Game Theory

The problem explored in this thesis involves finding the best trajectory for a penetrating UAV or other aircraft that travels deep inside the detection and engagement rings of modern SAMs in enemy territory to perform surveillance, take imagery, and return safely to friendly areas. Many researchers have studied trajectory optimization in great detail [19]. Methods used in the past include Mixed-Integer Linear Programming, which allows optimization applications in areas where variables must have integer values [14]. Probabilistic Road Maps work by combining large, pre-optimized routes with a series of small path segments to reach a goal [11]. Rapidly-exploring Random Trees uses only the small path segments to complete an entire route from start to end [12]. Voronoi diagrams utilize multiple connected edges positioned appropriately to allow for the integration smooth, flyable trajectories [13]. Studies using these methods have provided valuable insight, yet lack realistic scenarios or solutions. This thesis focuses on the specifics of penetrating a SAM ring, which necessitates analysis of other factors such as RCS management and image quality. In addition, any UAV path must take into account the capabilities and reach of the SAM. In a way, the UAV and SAM are engaging in a simple game. Specifically, they are engaging in a two person non-cooperative mixed zero-sum game. These terms are explained later in this chapter. The strategies of the UAV comprise the multitude of possible paths it could take to get a valid image. The strategies of the SAM include firing the missile at different times to shoot down the UAV. With the use of game theory, a two-sided game such as this one can be scrutinized and solved to provide the most likely outcome.

4.1 A Brief History of Game Theory

A scattering of economists and mathematicians anticipated many of the ideas of game theory throughout the 19th and early 20th centuries [18]. The economists Augustin Cournot and Francis Ysidro Edgeworth published notable papers on the competition between producers and trading between individuals, respectively [21]. In studying the game of chess in 1913, the mathematician E. Zermelo discovered the basis for backwards induction, otherwise known as Zermelo's Theorem [21]. Emile Borel presented the first modern formulation of a mixed strategy, among other ideas, in the 1920s, and the mathematician John von Neumann later provided the basis for game theory in a paper published in 1928 [3]. However, the groundbreaking text on game theory was a collaboration between von Neumann and Oskar Morgenstern called *The Theory of Games and Economic Behavior*, published in 1944 [4].

Development of game theory continued with John Nash's publication of four papers from 1950-1953 [5, 6] that made vital contributions to non-cooperative game theory and bargaining theory, including proving the existence of a strategic equilibrium (now termed the Nash equilibrium) for non-cooperative games [18]. The publication of A.W. Tucker's lecture on the Prisoners' Dilemma in 1950 made game theory even more widely known. Notable game theorists who continued to contribute throughout the latter half of the 20th century include Aumann [7], Shapley [8], Selten [9], and Harsanyi [10]. The year of 1994 brought a large amount of public interest in the subject when the Nobel prize in Economic Science was awarded to John Nash, John C. Harsanyi and Reinhard Selten for their contributions to game theory. The subject is one of increasing complexity, and it remains one of the most powerful tools available for studying human interaction.

4.2 Characteristics of Game Theory

Game theory describes interactions in which the outcomes depend on the individual strategies of two or more rational players whose motives are opposed or at least mixed [18]. Game theory has been applied in detail and practice to a variety of disciplines with a heavy emphasis on economics. For example, a person bidding at an auction is playing a game with the other bidders. A supermarket is playing a game with its customers and with other stores when it

decides to sell cereal at a certain price, and lawyers play games with each other when they decide on what type of defense or prosecution will be pursued [21]. Games can also be more serious in nature. Napoleon and Wellington were playing a game at the Battle of Waterloo, and Khrushchev and President Kennedy were playing a game during the Cuban missile crisis [21]. In reality, many situations in life can be described in terms of game theory.

4.2.1 Nomenclature

Some terms must first be introduced to appropriately define the UAV/SAM game. A dominant strategy is one that a player will choose regardless of the other players' strategies because it gives him maximum benefits regardless of the activities of others. If every player in a game utilizes a dominant strategy, then the outcome is called a dominant strategy equilibrium, which might also be a Nash equilibrium. However, a Nash equilibrium is not always a dominant strategy equilibrium. The Nash equilibrium applies only when a set of chosen strategies in a game exist such that no player can benefit by changing his strategy if the other players keep their strategies unchanged. What makes some games more interesting and problematic than others is the existence of multiple Nash equilibria [18].

Games also exist in many forms. Games can be played by more than two players, and in fact, the majority of interactions in the world occur between more than two parties. Some games are non-cooperative, although many games exist in which the goals of each party are the same and thus cooperation is possible to achieve an outcome. Also, the payoffs for each party in a game may sum to a constant (or zero), or it could be a non-constant sum game, which is more common in practical situations.

However, the losses and rewards are equal in a zero-sum (or constant-sum) game, which can be either pure or mixed. A pure zero-sum game is one in which each player chooses only one strategy. A mixed zero-sum game is one in which a player chooses among two or more strategies at random according to different probabilities. The solution to a two person zero-sum game is quite clear and is called the maximin strategy. This occurs when each player chooses the strategy (or set of strategies) that maximizes their minimum payoff.

The problem of routing a penetrating UAV inside the detection radius of a SAM to take imagery and return to safety can be modeled as a two person non-cooperative mixed zero-sum

game. The game is non-cooperative because the UAV and SAM have different goals. The problem is modeled as a zero-sum game for simplicity because the UAV's goals (take an image, complete mission safely, etc..) are exactly the opposite of the SAM's goals (prevent image acquisition, shoot down UAV, etc...). It will also be shown in later chapters that multiple Nash equilibria in the solution to this game will dictate mixed strategy solutions.

4.3 Calculating the Nash Equilibrium

Consider the payoff matrix $A = (a_{i,j})$ [20]. For a two person zero-sum game, one player chooses action i (out of n possible actions), and the other player chooses action j (out of m possible actions). Player one and player two receive payoffs of $a_{i,j}$ and $-a_{i,j}$, respectively, where (\hat{i}, \hat{j}) is defined as the outcome of the game. A dominant strategy i^* for player one occurs when $a_{i,j} \leq a_{i^*,j}$ for all i and j . Similarly, a dominant strategy j^* for player two occurs when $a_{i,j} \geq a_{i,j^*}$ [20]. For a mixed zero-sum game, the players have mixed strategies, meaning that player one will select $\mathbf{x}^T = (x_1, x_2, \dots, x_n)$ where x_i corresponds to the probability that player one will choose action i . Player two will select $\mathbf{y}^T = (y_1, y_2, \dots, y_m)$ where y_i corresponds to the probability that player two will choose action j [20].

The goal of this game is for player one to maximize his average payoff and for player two to minimize player one's payoff (for a zero-sum game). In other words, player one wants to do the following [20]:

$$\text{maximize } \mathbf{x}^T A \mathbf{y} = \sum_i^n \sum_j^m x_i a_{i,j} y_j. \quad (4.1)$$

Conversely, player two would like to attempt the following [20]:

$$\text{minimize } \mathbf{x}^T A \mathbf{y} = \sum_i^n \sum_j^m x_i a_{i,j} y_j. \quad (4.2)$$

In a zero-sum game, only one payoff matrix is needed, and a mutual gain or loss in terms of payoff is not possible [20].

For mixed zero-sum games and if $A \geq 0$, then an equilibrium strategy for player one solves $\max_x v$ subject to $v \leq \sum_i x_i a_{i,j}$ for all j , with $\sum_i x_i = 1$ and $x_i \geq 0$. The optimal value of v corresponds to the average payoff to player one in equilibrium. If $x'_i = x_i/v$, then the following

formula will equivalently find the Nash equilibrium:

$$\text{minimize } \sum_i^n x'_i \quad \text{subject to} \quad \left\{ \begin{array}{l} \sum_i x'_i a_{i,j} \geq 1, \quad \text{for all } j \\ x'_i \geq 0, \quad \text{for all } i \end{array} \right\}. \quad (4.3)$$

Similarly, $\mathbf{y} = \mathbf{y}' / \sum_j y'_j$ is the equilibrium strategy for player two if $\mathbf{y}' = (y'_1, \dots, y'_n)$ is a solution to:

$$\text{maximize } \sum_j^m y'_j \quad \text{subject to} \quad \left\{ \begin{array}{l} \sum_j a_{i,j} y'_j \leq 1, \quad \text{for all } i \\ y'_j \geq 0, \quad \text{for all } j \end{array} \right\}. \quad (4.4)$$

The payoff to player one is $v = 1 / \sum_i x'_i = 1 / \sum_j y'_j$. More information on game theory can be found in Appendix F.

4.4 UAV vs SAM

The calculation of the performance parameters I_q and P_k are designed to provide variables with which the positive and negative aspects of different strategies can be traded off. The first step is to combine the relevant parameters into an overall performance metric that can be placed inside the payoff matrix A . This process takes as inputs I_q , P_k , and completion time (t_c) for each UAV trajectory. It takes these values and uses a simple equation to combine them into a single metric:

$$A_{i,j} = (I_q - c \cdot t_c)_{i,j} \cdot (1 - P_k)_{i,j}, \quad (4.5)$$

where i and j are the UAV and SAM strategies, respectively, and c is a constant, normally set to 0.001. This value represents the weighting given to completion time. Thus, it is clear that completing the mission quickly is not necessarily the most important factor. If the UAV takes many hours to complete the mission but is able to avoid the enemy radar and obtain an excellent image, then that trajectory would be ranked quite high regardless. t_c is found by summing the distance between each waypoint, w_i , and dividing by the UAV velocity, v_i :

$$t_c = \sum_{i=1}^M \frac{(w_{i+1} - w_i)}{v_i}, \quad (4.6)$$

where M is the number of waypoints in each UAV path.

Other trends in Equation 4.5 are that as I_q gets higher, the payoff $A_{i,j}$ increases. Also,

as P_k gets higher, $A_{i,j}$ decreases. The UAV's goal in this game is to maximize $A_{i,j}$ whereas the SAM's goal is to minimize $A_{i,j}$. The Nash equilibrium is then found using Equations 4.3 and 4.4. The built-in Matlab function "linprog" finds the Nash equilibrium using the following form:

$$\begin{aligned} \min_x \quad & f^T x \quad \text{such that } A \cdot x \leq b \\ & A_{eq} \cdot x = b_{eq} \\ & lb \leq x \leq ub. \end{aligned} \tag{4.7}$$

A vector of ones whose length is determined by the number of possible UAV trajectories is used for f , the cost. A is simply the payoff matrix. b is also a vector of ones, which forces the resulting path probabilities to be less than one. The equality constraints on A and b are not used, and the bounds on x are set to zero (lb) and an arbitrary large positive number (ub). The outputs of this function are the optimization metric (x), the payoffs for the UAV and SAM ($\hat{A}_{i,j}$), and a vector of probabilities (p). This vector is essentially the solution, or Nash equilibrium, of the game. It gives the probability that each path should be chosen so that in the long run, the Nash equilibrium would be achieved.

One interesting feature of game theory is that unusual solutions sometimes emerge. The solution may be that only one specific path should be chosen with a 100% probability and the rest with a 0% probability. However, the solution could also be that a set of paths should be chosen with varying probabilities. This presents difficulties when attempting to select only one path for the UAV to fly and one strategy for the SAM. Thus, an element of chance must be introduced. Essentially, random numbers are generated, and the previously calculated probabilities are used to select a single UAV strategy and a single SAM strategy. If this process was repeated a large amount of times, the frequency of strategies chosen would converge to the initial probability vector. The resulting P_k and I_q are tabulated for use in later Monte Carlo simulations. All of the computer code used in the simulation is located in Appendix G.

Regarding the sample UAV trajectory discussed previously, it was not chosen as part of the Nash equilibrium for that specific run, in spite of its 6.04 I_q and most likely because of its high P_k values. Instead, the path in Figure 4-1 was chosen with a probability of 100% due to its

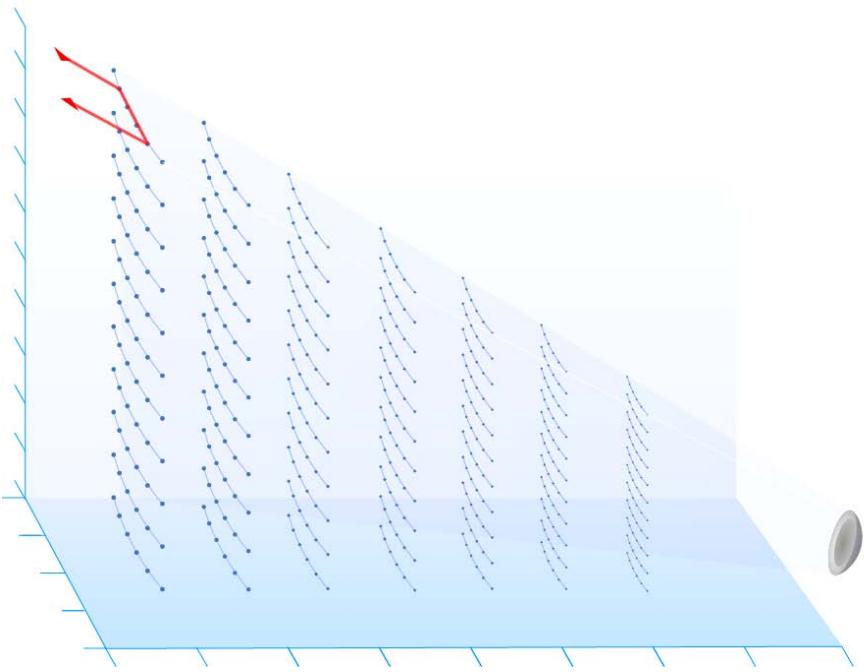


Figure 4-1: UAV Path given by Nash Equilibrium. This path was chosen with a probability of 100% due to its extremely low probabilities of kill (0.01 for each SAM strategy) even though the best possible image would be only 4.0894. Thus, the safe trajectory won out in this example.

extremely low probabilities of kill (0.01 for each SAM strategy) even though the best possible image would be only 4.0894 on the NIIRS level. Evidently, the strategy of a safe, less risky trajectory won out in this particular example.

Chapter 5

Results and Conclusions

Four separate situations with increasing complexity were analyzed. These situations imply different amounts of knowledge about the enemy environment as well as differing strategies about how to deal with the situation. In the first case, the SAM's location is assumed to be known and fixed. The second case encompasses the first one but introduces uncertainty in the location of the SAM site. It is assumed that the UAV is not "smart" and thus does not take the uncertainty into account. The third case builds on the second case and assumes that the UAV has now increased its knowledge of the SAM and will act more intelligently. Specifically, the SAM is assumed to be at one of a number of possible SAM locations. The fourth case represents the most realistic scenario. A wide FOV asset such as ELINT or DSP will pass a cue to the penetrating UAV, which must then search an area for the target. These four cases embody the full spectrum of uncertainty in SAM location and knowledge and provide interesting results and conclusions. The results presented here, although accurate and straightforward, must be analyzed carefully and in proportion to the quality of the model. Unfortunately, many assumptions and estimates were used, which lowers the fidelity of the model. However, these results do provide an interesting perspective on the dangers of air defense threats in the 21st century.

Test Cases	SAM Radius (km)	UAV Speed (knots)	Missile Speed (Mach)
1	50	150	3
2	75	150	3
3	100	150	3
4	50	350	3
5	50	550	3
6	50	150	6
7	50	150	9

Table 5.1: Input Parameter Values for Case One Test Cases

5.1 Case One

Case one consists of the following conditions:

$$s \text{ known}, s \in \{\tilde{s} : \|\tilde{s} - s\| = 0\}. \quad (5.1)$$

The variable s is where the UAV believes the SAM is located. \tilde{s} is where the SAM is actually located. The difference between those two values is zero in Equation 5.1, meaning that the SAM's location is known exactly by the UAV. The maximin strategy will be used to enable the UAV to find an optimal route that maximizes its payoff, $A(s)$, based on that known location, s .

$$\max_x \min_y \{\underline{x}^T \cdot A(s) \cdot \underline{y}\}. \quad (5.2)$$

This type of specific information may have come from the predictive archives. This case is probably the most unrealistic due to the high uncertainty in SAM locations. However, it does provide a valuable baseline for comparison with other cases.

The computer simulation was run 10 times each (due to run time constraints) for a variety of test cases with each test case using different inputs. It was determined that the three user-defined input parameters that could realistically change and have the most impact on the resulting equilibrium would be the detection radius of the SAM, the speed of the UAV, and the speed of the actual missile. Table 5.1 lists the specific values of these parameters for each of the seven test cases.

The computer simulation was run with a SAM detection radius of 50, 75, and 100 km. It was also run at UAV speeds of 150, 350, and 550 knots as well as at missile speeds of Mach 3,

Mach 6, and Mach 9. The resulting P_k and I_q were averaged over each of the 10 runs for every test case.

The results showed that increasing the detection radius had a negligible effect on the P_k of the UAV. This is probably due to the fact that no matter how large the detection radius is, the UAV will always be able to penetrate the radius to some extent as long as the exact SAM location is known, allowing the UAV to manage its RCS. However, the increase in radius did have a slight effect on the I_q , illustrated in Figure 5-1.

Thus, the UAV will always find the optimal route that minimizes P_k , but that path will yield slowly deteriorating image quality as the detection range increases.

Test cases one, four, and five outline the influence of UAV speed in the simulation, which gave interesting results. One might think that increasing the speed of the UAV would allow it to complete the mission faster and spend less time inside the SAM's detection radius, which would then decrease the P_k . However, the results of this simulation demonstrate that the opposite is true. Figure 5-2 shows that as UAV speed is increased, P_k increases significantly. One reason for this increase could be that a fast UAV would be more likely to show up on a Doppler radar. Essentially, its speed would give it away whereas a slower UAV uses its lack of speed to its advantage by maintaining a small Doppler velocity and thus lowering its chances of being detected. Also, variation in UAV speed had no effect on I_q .

An analysis of test cases one, six, and seven demonstrated that missile speed had no effect on either P_k or I_q . The slowest simulated speed of the missile (Mach 3) is already so large that the missile will reach the UAV in mere seconds. Thus, increasing that speed to Mach 6 or Mach 9 will do little to increase the P_k , and it certainly will not have any effect on I_q .

Several comments can be made about the simulation in general. The UAV trajectories picked by the Nash equilibrium utilized higher altitudes, most likely to gain better depression angles from which to obtain an image. Although some paths took the UAV deep into the SAM radius, the majority of the paths chosen stayed toward the edge of the SAM radius, picking safety over image quality. Also, the paths that utilized zigzagged or continuous beaming maneuvers were chosen less frequently than those that led directly in and out. This result suggests that in seeking to minimize radar detection, the UAV found it best to simply minimize its time inside the SAM radius rather than lengthen that time by performing more complex

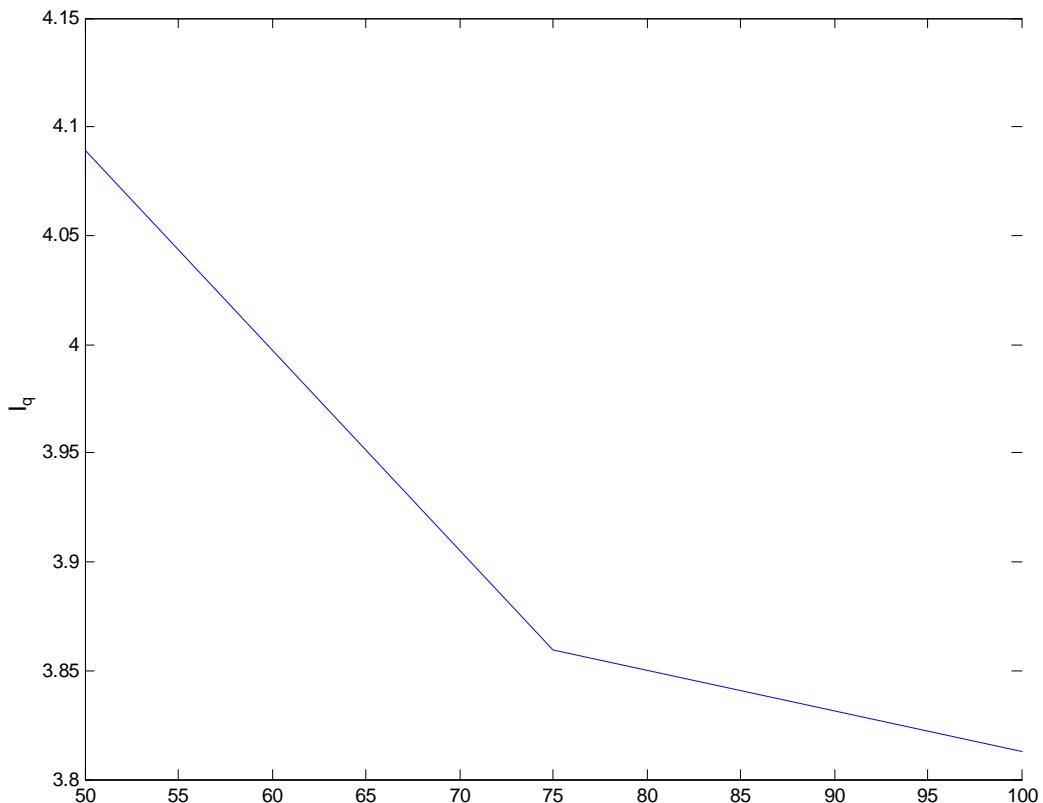


Figure 5-1: Image Quality as Detection Radius Varies. As the detection radius of the SAM increases from 50 to 100 kilometers, the NIIRS level of the best possible image taken by a penetrating UAV decreases slightly from about 4.1 to about 3.8 with a missile speed of Mach 3.

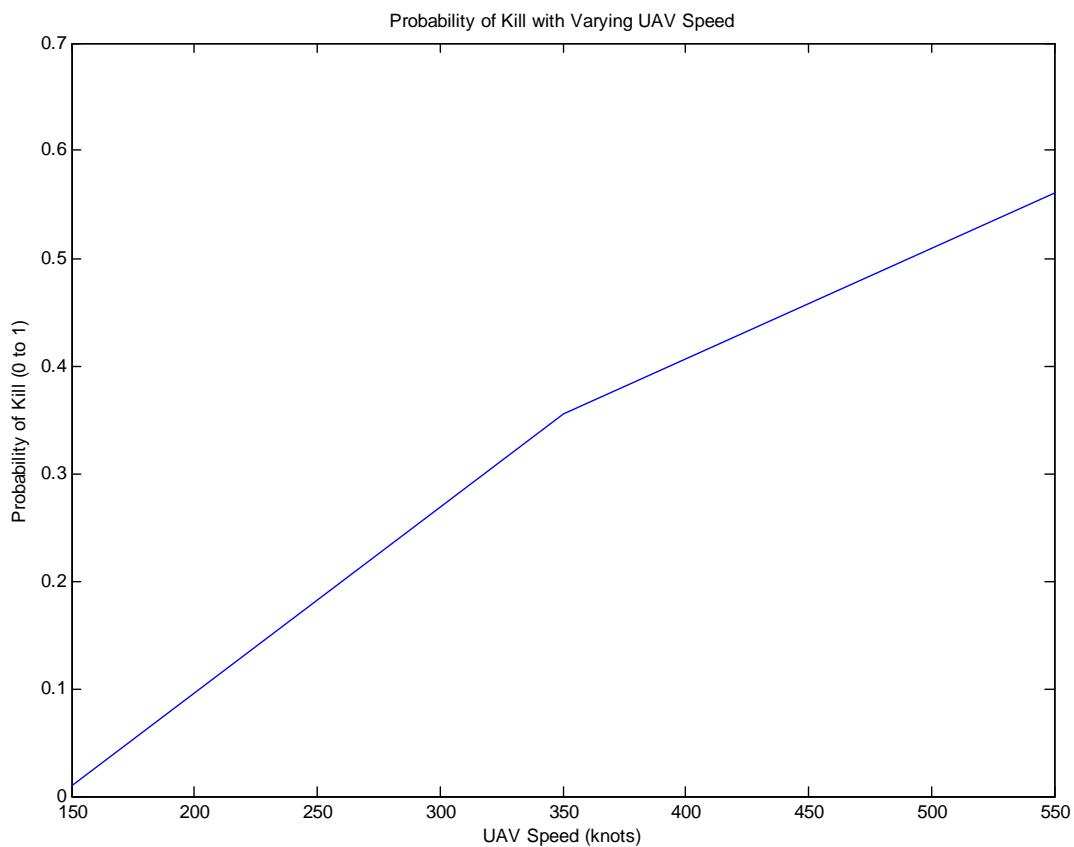


Figure 5-2: Probability of Kill with Varying UAV Speed. Clearly, the UAV's probability of kill increases as the UAV's speed increases. Specifically, the UAV's speed was simulated at 150, 350, and 550 knots. The resultant probability of kill went from 0.01 to about 0.35 to about 0.55.

and risky maneuvers. The SAM strategy picked most frequently by the Nash equilibrium was the first one, which gave the highest values of P_k . This strategy was the one in which the SAM fired upon the UAV at first detection. Evidently, the second strategy in which the exiting UAV was fired upon gave too much time to the UAV and thus lowered the average values of P_k . With regard to the game-theoretic component of the simulation, the optimization terminated successfully on every occasion, yielding results that made sense analytically as well as numerically.

5.2 Case Two

Case two brings more reality to the situation by introducing some uncertainty in the location of the SAM. In an actual engagement with the enemy, the exact location of a particular SAM site is rarely known. The predictive archives might suggest a SAM location with some uncertainty. Thus, it would be useful to determine what effect the uncertainty in location has on the P_k and I_q . Using the input parameters of test case one, the simulation was run with increasing magnitude of target location uncertainty. It was assumed in this case that the UAV is not "smart" and thus does not take the uncertainty into account. It simply calculates and flies its optimal route as in the previous case thinking that the SAM is still at its initial location. This case is described mathematically using the formula in Equation 5.3.

$$s \in \{\tilde{s} : \|\tilde{s} - s\| \leq \delta\} = N(s, \delta), \quad (5.3)$$

\tilde{s} , the actual SAM location, is really within some radius, δ , which represents the amount of uncertainty in the SAM location. The set of different SAM locations within δ is termed $N(s, \delta)$ where N is a function of where the UAV thinks the SAM is, s , and how much uncertainty exists in the SAM's location, δ . The optimization routine is modified to automatically increase δ .

$$\frac{1}{|N(s, \delta)|} \sum_{\tilde{s} \in N(s, \delta)} \max_{\underline{x}} \min_{\underline{y}} \{\underline{x}^T A(\tilde{s}) \underline{y}\}. \quad (5.4)$$

In this case, the maximin strategy is again used. However, the payoff matrix that is optimized is $A(\tilde{s})$, which includes the uncertainty in SAM locations. Additionally, the payoffs are summed

over the entire set of possible SAM locations and then divided by the magnitude of the set to give the average payoff for a specified value of uncertainty. The simulation was run for an uncertainty of up to 10 miles with a step size of 0.1 miles. At each 0.1 mile increment, 1000 iterations were performed and averaged to give a P_k and I_q for an uncertainty of that magnitude. The graph in Figure 5-3 is the result of that run for P_k . When the uncertainty in SAM location is only 0.5 miles, the P_k has jumped from 0.01 to about 0.56. After about the four-mile mark, the P_k essentially steadies out at 0.99. This literally means that if the uncertainty in the SAM location is only four miles or greater, the penetrating UAV flying its optimal route is virtually guaranteed to get shot down. The quality of the best image that can be taken by the SAM also degrades slightly as uncertainty increases. This result is shown in Figure 5-4. The specific sensor package with which each UAV is equipped necessitates a certain look angle and viewpoint from which the image must be taken. If the SAM happens to be at an unexpected location, the UAV will have a more difficult time obtaining a quality image. In this case, the average I_q was 3.89 at an uncertainty of 10 miles compared to 4.1 with no uncertainty.

Another run was performed which mapped the uncertainty out to 100 miles. For this run, the step size was one mile, and 1000 iterations were performed at each one-mile increment. The result for P_k is shown in Figure 5-5. Clearly, the SAM's denial of UAV penetration continues until the uncertainty rises to about 30 miles, at which point P_k steadily decreases to about 0.75 at 100 miles. This result is certainly not intuitive. One would expect the SAM to be able to shoot down all UAVs as the uncertainty in its location gets larger and larger. However, the actual reason is very simple. After uncertainty increases to such an extent, the UAV's optimized path and special maneuvers cease to have an effect on its P_k . The only factor affecting the ability of the SAM to shoot down the UAV is the range of the SAM's missile. Thus, at an uncertainty of 100 miles, the only time that a SAM anywhere within that limit was unable to shoot down a UAV was when the UAV was greater than 100 miles away from the SAM. This idea is explained visually in Figure 5-6. The blue area at the top represents the possible SAM locations that are outside the range of the UAV. The red area in the middle represents the possible SAM locations that are able to shoot down the UAV (that are within the uncertainty), and the gray area at the bottom represents points outside of the SAM's location uncertainty

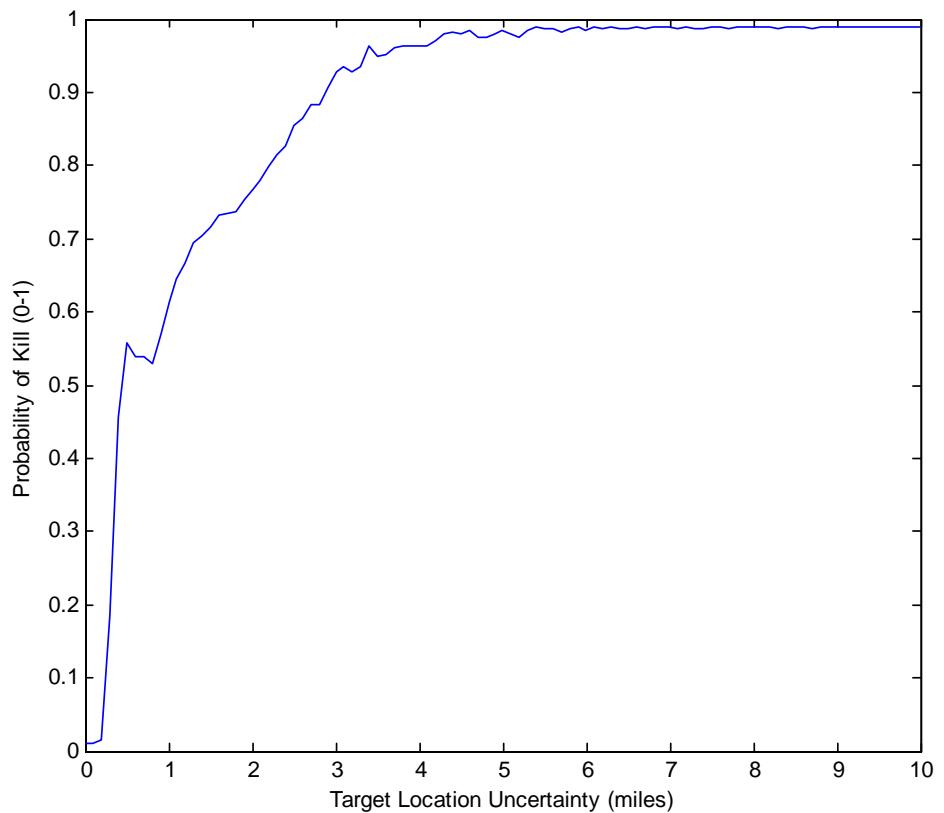


Figure 5-3: Probability of Kill with Zero to 10 miles of Uncertainty. When the UAV knows the SAM location exactly, the probability of kill is at 0.01, essentially zero. However, when the uncertainty in location rises just a little bit to 0.5 miles, the probability of kill jumps to 0.55. Also, after about four miles of uncertainty, the UAV's probability of kill steadies out at 0.99, assuring the destruction of any UAV.

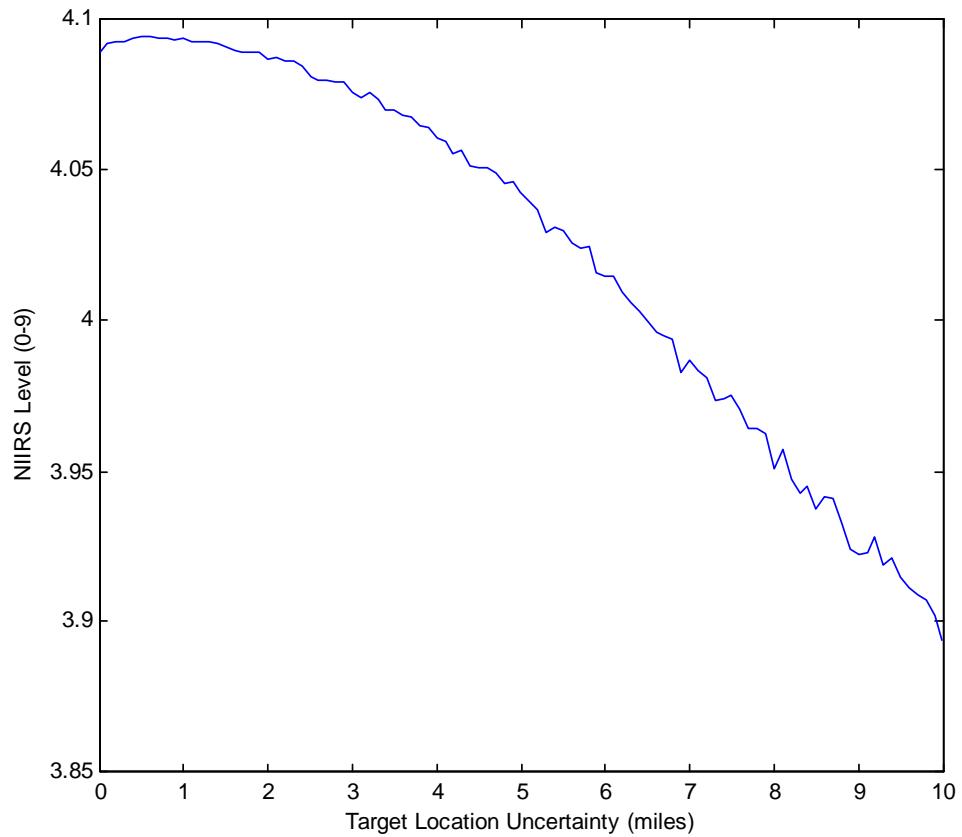


Figure 5-4: NIIRS Level with Zero to 10 mile Uncertainty. Initially, with no uncertainty, the UAV is able to obtain an image with a NIIRS level quality of about 4.1. After 10 miles of uncertainty, that level has decreased to about 3.9.

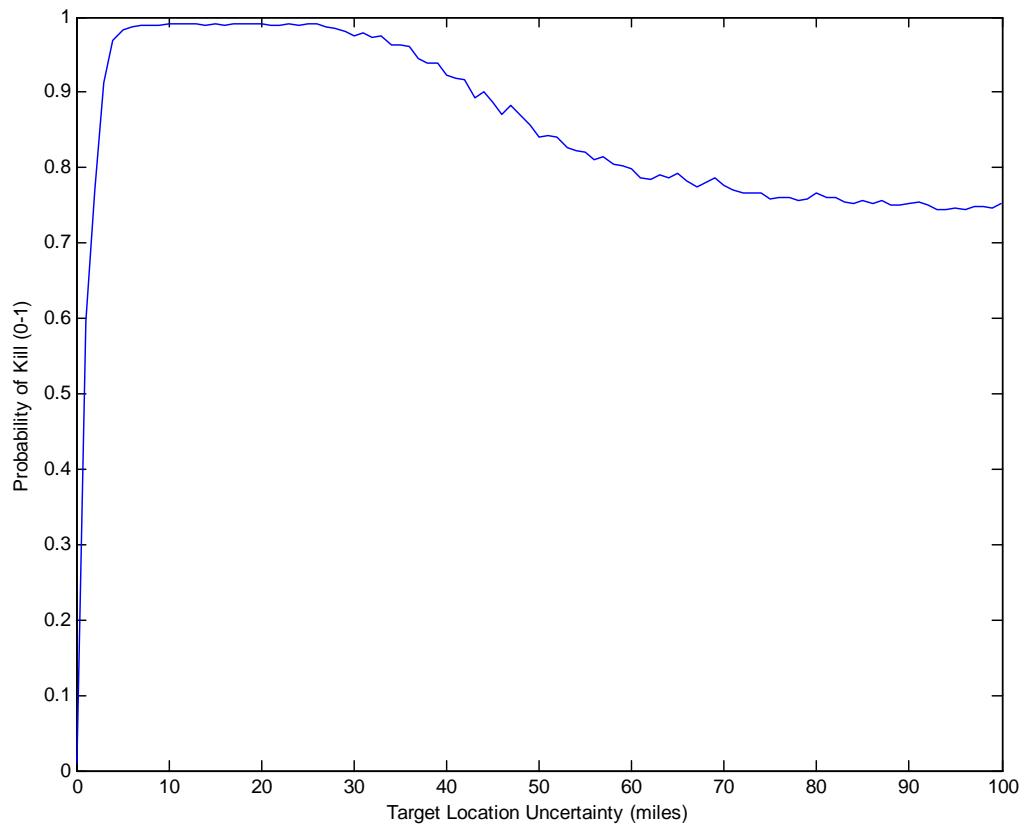


Figure 5-5: Probability of Kill with Zero to 100 mile Uncertainty. This graph plots the uncertainty out to 100 miles. The UAV is virtually guaranteed to get shot down until after 30 miles of uncertainty, at which point the probability of kill decreases to about 0.75. This value roughly corresponds to the proportion of times that the SAM is simply out of range of the UAV.

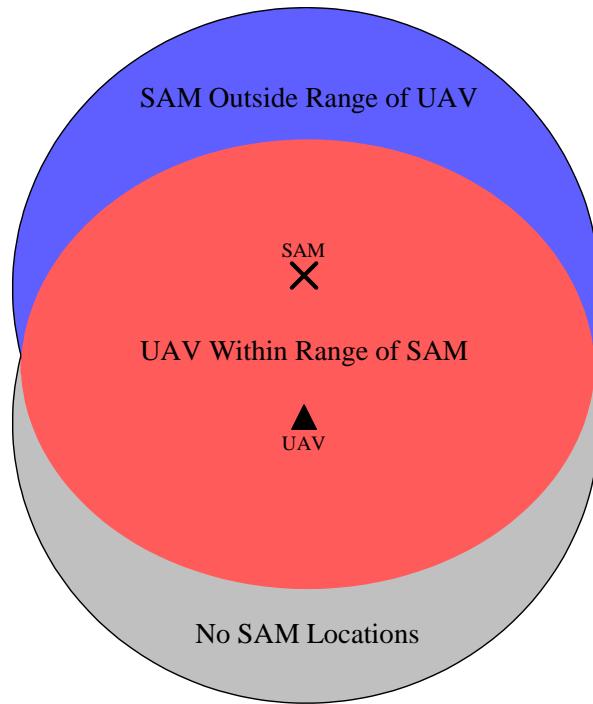


Figure 5-6: SAM Engagement and Location Uncertainty Areas. The blue area at the top represents the possible SAM locations that are outside the range of the UAV. The red area in the middle represents the possible SAM locations that are able to shoot down the UAV (that are within the uncertainty), and the gray area at the bottom represents points outside of the SAM's location uncertainty (but within the kill range of the UAV).

(but within the kill range of the UAV). The following equation was used to calculate the red area, which is the area of overlap between the circle of the SAM's possible locations (the blue circle) and the circle describing the area from which the UAV could be shot down (the gray circle) [15]:

$$\begin{aligned} A &= r^2 \arccos\left(\frac{d^2 + r^2 - R^2}{2dr}\right) + R^2 \arccos\left(\frac{d^2 + R^2 - r^2}{2dR}\right) - c \\ c &= \frac{1}{2} \sqrt{(-d + r + R)(d + r - R)(d - r + R)(d + r + R)}, \end{aligned} \quad (5.5)$$

where r is the radius of the gray circle, R is the radius of the blue circle, and d is the distance between the centers of the two circles (the distance between the SAM and UAV). Using the point of 100 mile uncertainty, where $r = 100$ miles, $R = 100$ miles, and $d = 50$ miles, Equation 5.5 gives $21,521.09$ km 2 . Dividing that number by the total possible area of SAM locations (the blue circle), which equals $\pi R^2 = 31,415.93$ km 2 , gives a percentage of approximately 0.69. This means that about 69% of the time, the SAM is out of range of the UAV for a 100 mile value of uncertainty. This number roughly corresponds with the 0.75 P_k shown in Figure 5-5 for a 100 mile uncertainty. The small difference between the two numbers may be due to the fact that in Figure 5-6, the UAV's path was approximated as a single point. Thus, as target location uncertainty gets very large, the UAV will be shot down every time except when it is outside of the range of the SAM.

Finally, the result of this run for I_q is shown in Figure 5-7. The decreasing trend of image quality is even more prominent when I_q is mapped out to 100 miles, showing an average NIIRS level of about 0.5 at that point.

5.3 Case Three

Case three builds on case two in that it assumes the UAV has now increased its knowledge of the enemy and will act more intelligently. Typical battlefield intelligence from the predictive archives or elsewhere may point to possible locations for a SAM site. In the first scenario, three specific locations (approximately tens of kilometers apart) are defined with associated probabilities that the SAM will actually be there. A table of these locations and their associated

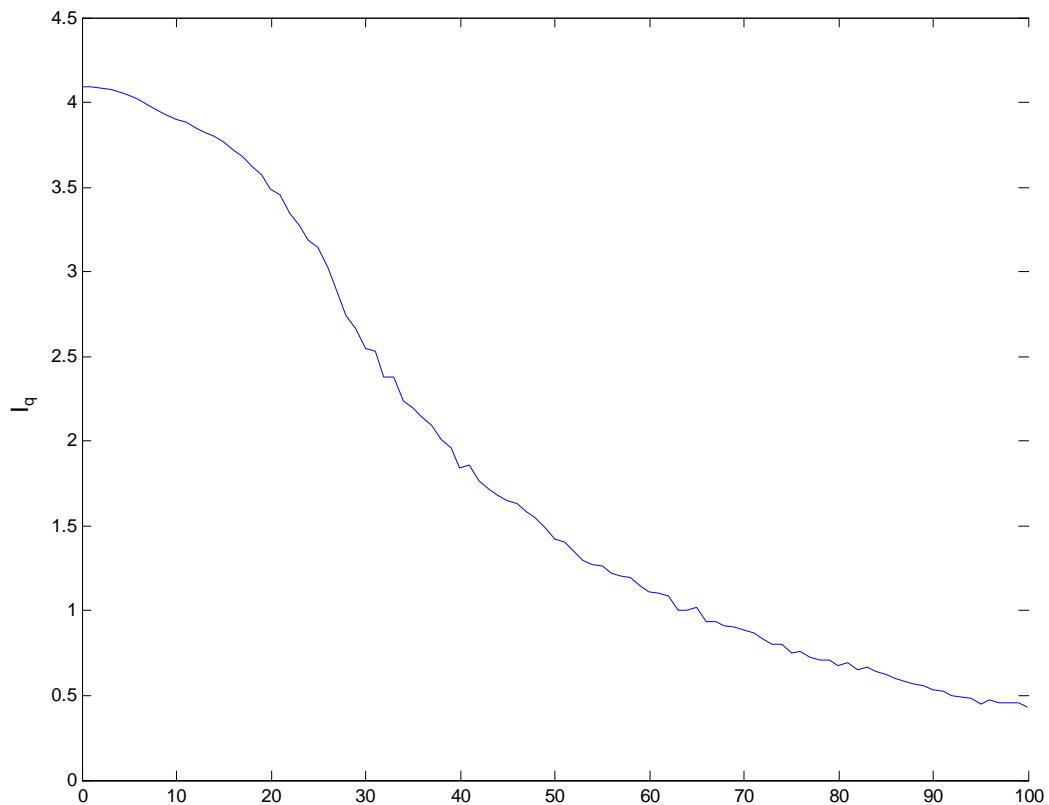


Figure 5-7: NIIRS Level with Zero to 100 mile Uncertainty. The image quality, embodied by the NIIRS level, decreases significantly after 100 miles of target location uncertainty from a possible 4.1 to a very low 0.5.

SAM Locations	Latitude (deg)	Longitude (deg)	Probability
1	38.85	126.85	0.5
2	38.45	126.45	0.3
3	39.25	127.25	0.2

Table 5.2: SAM Locations for Scenario One in Case Three

SAM Locations	P_k	I_q
1	0.01	4.0894
2	0.48	4.0762
3	0.99	4.3332

Table 5.3: Case Three Results for Scenario One using Average Payoffs

probabilities is shown in Table 5.2. The UAV will now attempt to use the known probabilities and locations to optimize its flight path according to the following formula:

$$s \in \{s_1, s_2, \dots, s_N\}. \quad (5.6)$$

The possible SAM locations, $\{s_1, s_2, \dots, s_N\}$, are known by both the UAV and the SAM. N represents the number of possible locations. The same maximin optimization strategy is used in Equation 5.7.

$$\max_{\underline{x}} \min_{\underline{y}} \left\{ \underline{x}^T \left[\sum_i A(s_i) \cdot p_i \right] \underline{y} \right\}. \quad (5.7)$$

In this case, each payoff matrix, $A(s_i)$, is multiplied by its associated probability p_i . All of the payoff matrices are summed, and the Nash equilibrium is found for that average payoff matrix, giving the best possible trajectory given the possible SAM locations and their probabilities. This case is much more realistic than case one, in which the SAM location was known with absolute certainty. Before using the formula in Equation 5.7, the program was run separately with the nominal set of input parameters for each SAM location in 5.2, simulating that the UAV knew each time with 100% probability that the SAM was going to be there. The obvious result was that the P_k was 0.01 for each location. However, the simulation was then run using the updated probabilities and the same input parameters. The result was an average P_k of 0.49 and an average I_q of 4.166. Thus, the UAV's chance of success is much less certain. The specific results for each location are shown in Table 5.3. This means that if the SAM was actually at

SAM Locations	P_k	I_q
1	0.7955	6.1237
2	0.7778	6.1681
3	0.7907	5.6075

Table 5.4: Case Three Results for Scenario One using Minimum Payoffs

location three, the UAV would have a 99% chance of getting shot down. At location two, it would have a 48% chance. It is clear from the results in 5.3 that the Nash equilibrium produced a solution that heavily weighted location one because of its high associated probability.

Although the P_k did increase somewhat, the UAV does have greater than a 50% chance of surviving the engagement and obtaining a picture in scenario one. One lesson may be that using Equation 5.7 leads to misleading results. However, another strategy states that one should always plan for the worst case scenario. Thus, the optimization formula could be changed slightly:

$$\max_{\underline{x}} \min_{\underline{y}} \left\{ \underline{x}^T \min [J(s_i)] \underline{y} \right\}. \quad (5.8)$$

The difference in Equation 5.8 is that instead of using the average, the minimum over all of the payoff matrices is found, which might possibly give a more realistic approach to the path planning. This strategy resulted in expected higher values of P_k , with an average of 0.79 and each individual location result shown in Table 5.4. Thus, using the minimization technique effectively took away the extreme 0.99 and 0.01 P_k values in Table 5.3 but caused the average P_k value to increase by 0.3. Interestingly, I_q increased as well, but the higher P_k would still not be worth obtaining the image.

The previous scenario involved only three possible locations. To provide some perspective on what would occur if the number of locations increased, the simulation was run for five, seven, and nine points as well, whose locations are shown in Table 5.5. The minimization formula in Equation 5.8 was used to simulate the worst case. The result for average P_k is shown in Figure 5-8. Clearly, as the number of SAM locations increase, the average P_k increases quickly, rising to 99% with seven possible locations. The average I_q remained fairly constant throughout the increase in locations, which could be due to the fact that all of the SAM locations were roughly in the same geographical area. Thus, poor images from extreme points would be balanced by excellent images from close points depending on the specific route taken through that area. A

SAM Locations	Latitude (deg)	Longitude (deg)
1	38.85	126.85
2	38.45	126.45
3	39.25	127.25
4	38.65	126.65
5	39.05	127.05
6	38.25	126.25
7	38.95	126.95
8	39.15	127.15
9	38.55	126.55

Table 5.5: SAM Locations for Scenario Two in Case Three

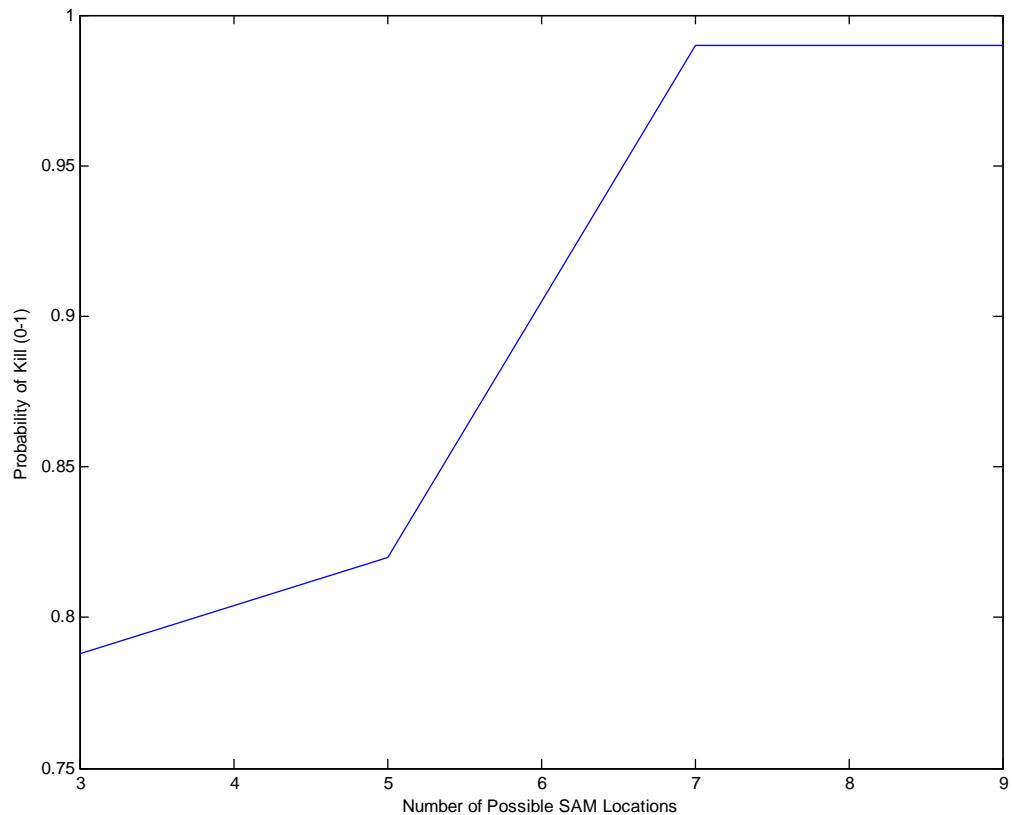


Figure 5-8: Probability of Kill as Possible SAM Locations Increase. When dealing with three possible SAM locations, the UAV can at best attain a 0.79 probability of kill on average. However, when the number of possible SAM locations rises to seven, the UAV is virtually guaranteed to get shot down no matter what techniques are used for this example.

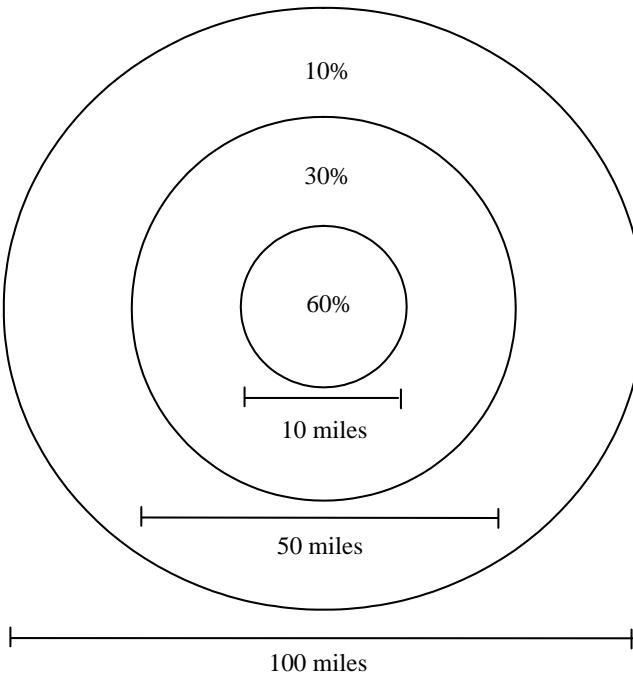


Figure 5-9: Scenario Depiction for Case Four. The typical wide FOV elliptical search area is defined here as a circle for simplicity. The outer ring is 100 miles in diameter, and there is a 10% probability that the SAM is located there. The middle ring is 50 miles in diameter with an associated 30% probability. The center ring is 10 miles in diameter with an associated probability of 60%. Given these regions and probabilities, the UAV must optimize its path to obtain the best picture and minimize its probability of kill.

future consideration not analyzed here might be the spacing of the points and how that affects the P_k .

5.4 Case Four

The fourth and final case analyzed here probably represents the most likely battlefield scenario. In this case, a wide FOV asset passes a cue to the penetrating UAV, which must then search an area for the target. That area is usually defined by an ellipse. However, the area will be depicted as a circle here for simplicity. Figure 5-9 shows a sample scenario for which the probabilities of the SAM are located in each area along with the area's associated dimensions. The UAV must again optimize its path through the area. The relevant condition on this

Area	Average Payoff - P_k	Minimum Payoff - P_k	I_q (Avg/Min)
Inner Ring	0.85	0.99	4.0720
Middle Ring	0.99	0.99	3.6727
Outer Ring	0.99	0.99	2.0716

Table 5.6: Case Four Results

optimization is as follows:

$$s \text{ continuous in } N(s, \delta). \quad (5.9)$$

The set of possible SAM locations, $N(s, \delta)$, is now continuous within a certain region, meaning that the SAM can now be located anywhere in that region. The optimization formula is:

$$\max_{\underline{x}} \min_{\underline{y}} \left\{ \underline{x}^T \left(\min_i A(s_i) \right) \underline{y} \right\}. \quad (5.10)$$

This equation is very similar to that of case three. The difference in this case is that the possible SAM location, s , is assumed to be continuous throughout the regions in Figure 5-9. The optimization routine was run using the average payoff strategy as well as the worst case minimization strategy depicted in Equation 5.10. The continuous SAM locations were modeled by running 1000 iterations for each mile of area radius. The results for this run are shown in Table 5.6, where the inner ring represents the 10-mile diameter area with a 60% probability, the middle ring is the 25-mile diameter area with a 30% probability, and the outer ring is the 50-mile diameter area with a 10% probability. Clearly, the UAV does extremely poorly even with knowledge of the SAM location probabilities. If the SAM happened to be within the inner ring, the UAV would get shot down 85% of the time. Aside from that occurrence, the UAV will get shot down every time with a probability of 99%. Also, as expected, the image quality decreases to approximately a NIIRS level two from a maximum of level four.

5.5 Conclusions

Finding optimal trajectories for penetrating UAVs in ISR missions is a difficult challenge. A variety of factors, including SAM sites and terrain obscuration, combine to decrease the quality of image attained and increase the probability of getting shot down. Specifically, it was shown using game-theoretic principles that an increase in the detection radius of a SAM site will cause

a decrease in I_q . Also, an increase in the speed of the UAV will cause an increase in P_k . Other characteristics of the UAV or SAM have negligible effects on both I_q and P_k . However, the one factor that has the most influence on an optimal trajectory is the uncertainty in the location of the SAM. After only 0.5 miles of uncertainty, the P_k of a UAV assuming perfect knowledge of the SAM location has already risen to 0.56. When the uncertainty rises to about four miles, the UAV is virtually guaranteed to get shot down with a probability of 0.99. After the SAM location uncertainty reaches 30 miles, the P_k starts to decrease to a value of about 0.75 at 100 miles of uncertainty, which roughly corresponds to the amount of time the UAV would be within range of the SAM. As expected, I_q decreases significantly as the uncertainty increases.

When the UAV takes uncertainty into account, the results are not encouraging. A situation was created with three possible SAM locations with varied probabilities that the SAM will actually be at those locations. Using the method of optimizing average payoffs results in an average P_k of 0.49 and an average I_q of 4.166. Using instead the worst case strategy of optimizing the minimum payoffs results in an average P_k of 0.79, which is more realistic. Extending this situation to five, seven, and nine possible SAM locations with varied probabilities resulted in an increase in P_k to 0.99 at seven locations. Thus, with seven or more different SAM locations, the UAV is 99% likely to get shot down using an optimal path. Using an even more realistic scenario, the UAV must then optimize its path throughout an area of varying probabilities. In this specific case, using the average payoff strategy, the UAV has an 85% chance of getting shot down if the SAM is located within a five mile radius of the center of the area. Outside of this area, the UAV is guaranteed to get shot down with a P_k of 0.99.

The results presented here certainly paint a dark picture of the survivability of penetrating ISR missions for UAVs or other aircraft given modern air defense threats when their locations are assumed to be known. Game-theoretic optimization of different vehicle trajectories will work only if the air defense locations are known, but even a slight variability in those locations will ruin the most perfectly planned path. It is often stated that dynamically tasking air vehicles to respond to observed changes in air defense locations will compensate for imperfect knowledge. However, with typical detection ranges of 100 miles, most air assets will more likely be shot down before detecting a SAM, whose engagement range may now extend past 400 km in some cases. Other techniques and methods must be explored and used in combination

with RCS management to ensure the continuing collection of valuable ISR imagery given the extended capabilities of modern air defense threats.

5.6 Future Research

More work can certainly be done to improve the accuracy and fidelity of the UAV strategies, the SAM strategies, and the game theoretic techniques. Specifically, a wider and more appropriate range of UAV strategies could be used, which might give the UAV a better chance at surviving engagements under uncertainty. The target detection and tracking process could be enhanced, and other SAM strategies could be added. Additionally, a larger variety of test cases and specific scenarios within those cases could provide more accurate results. Finally, exploration of better methods to mask aircraft from radar or research into new techniques to collect ISR imagery could prove invaluable in the future.

Bibliography

- [1] Pietrucha, M. "A Case for UCAV." Group 61, MIT Lincoln Laboratory. 15 September 2004.
- [2] "Combat Aircraft Fundamentals, Fighter Fundamentals." F-15E. MCM 3-3. Department of the Air Force. June 1994.
- [3] Von Neumann, J. "Zur Theorie der Gesellschaftsspiele." Mathematische Annalen, 1928.
- [4] Von Neumann, J. and Morgenstern, O. "Theory of Games and Economic Behavior" Princeton University Press (1944, 1947, 1953).
- [5] Nash, J.F. "The Bargaining Problem." Econometrica, 1950.
- [6] Nash, J.F. "Non-Cooperative Games." Annals of Mathematics, 1951.
- [7] Aumann, R.J. "Markets with a continuum of traders." Econometrica, 1964.
- [8] [CITATION] A value for n-person games LS Shapley - Cited by 487 - Web Search Contributions to the Theory of Games, Princeton University Press, 1953
- [9] CITATION] Games with Randomly Disturbed Payoffs: A New Rationale for Mixed Strategy Equilibrium Points J Harsanyi, R Selten - Cited by 163 - Web Search International Journal of Game Theory, 1973
- [10] [CITATION] Games with Incomplete Information Played by Bayesian Players JC Harsanyi - Cited by 261 - Web Search Management Science, 1967
- [11] Kavraki, L. et al, "Probabilistic Roadmaps for Path Planning in High-Dimensional Configuration Spaces." Technical Report CS-TR-94-1519, 1994.

- [12] Karatas, T. and Bullo, F. "Randomized Searches and Nonlinear Programming in Trajectory Planning." *Proceedings of the IEEE Conference on Decision and Control*. 2001.
- [13] McClain, T.W. and Judd, K.B. "Spline Based Path Planning for Unmanned Aerial Vehicles." *Proceedings of the AIAA Guidance, Navigation and Control Conference*. August 2001.
- [14] Williams, H.P. and Brailsford, S.C. "Computational Logic and Integer Programming." *Advances in Linear and Integer Programming*. Pages 249-281. Clarendon Press, 1996.
- [15] Weisstein, E.W. "Circle-Circle Intersection." From *MathWorld-A Wolfram Web Resource*. <<http://mathworld.wolfram.com/Circle-CircleIntersection.html>> Accessed 19 March 2005.
- [16] "Global Hawk Integrated Sensor Suite." Electronic Systems. Raytheon. 2000.
- [17] Humerfelt, S. "How WGS 84 defines the Earth." <http://home.online.no/~sigurdhu/WGS84_Eng> Accessed June 2004.
- [18] McCain, R.A. *Game Theory: A Nontechnical Introduction to the Analysis of Strategy*. South-Western Educational Publishing: New York, 2003.
- [19] Bellingham, J.S. "Coordination and Control of UAV Fleets using Mixed-Integer Linear Programming." Master's Thesis. Department of Aeronautics and Astronautics, Massachusetts Institute of Technology. September 2002.
- [20] Zwillinger, D. *CRC Standard Mathematical Tables and Formulae, 31st Edition*. Chapman and Hall/CRC Press LLC: Boca Raton, 2003.
- [21] Binmore, K. *Fun and Games, A Text on Game Theory*. D.C. Heath and Company: Lexington, 1992.
- [22] Leachtenauer, J.C. et al. "General Image-Quality Equation: GIQE." *Applied Optics*. Vol 36, No 32. 10 November 1997
- [23] Irvine, J.M. "National Imagery Interpretability Rating Scale (NIIRS)." *Encyclopedia of Optical Engineering*. 2002.

- [24] "What is Synthetic Aperture Radar?" Sandia National Laboratories. <<http://www.sandia.gov/RADAR/whatis.html>> Accessed 10 March 2005.
- [25] Anderson, J. et al. "UAV Vehicle and Sensor Characteristics." MIT Lincoln Laboratory Briefing Slides. 5 December 2002.
- [26] "Unmanned Aerial Vehicles Roadmap, 2002-2027." Office of the Secretary of Defense. December 2002.
- [27] Scott, J. "Radar Cross Section." <www.aerospaceweb.org> Accessed 8 March 2005.
- [28] "The Doppler Effect." Science and Industry Expo. Whispers from the Cosmos. University of Illinois. <<http://archive.ncsa.uiuc.edu/Cyberia/Bima/doppler.html>> Accessed 7 March 2005.
- [29] Electromagnetic Spectrum Graphic. Canada Center for Remote Sensing. <<http://www.ccrs.nrcan.gc.ca/ccrs>> Accessed 3 March 2005.
- [30] O'Donnell, R.M. "Introduction to Radar Systems." Lecture Slides. MIT Lincoln Laboratory. 7 January 2002.
- [31] Skolnik, M.I. *Introduction to Radar Systems, Third Edition.* McGraw-Hill: New York, 2001.
- [32] Lethbridge, Cliff. "History of Rocketry, Chapter 4." <www.spaceline.org> Accessed 5 March 2005.
- [33] "History of Radar." Physics, Science and Technology. BBC. <<http://www.bbc.co.uk/dna/ww2/A591545>> Accessed 2 March 2005.
- [34] "Anti-Air Weapons." <www.periscope.com> Accessed 2 March 2005.
- [35] "History of Radar." Weather, Radar. The Franklin Institute. <www.fi.edu> Accessed 5 March 2005.
- [36] Meiners, K. "UAVs in a Net-Centric World." Briefing Slides. 12 June 2003.

- [37] Pietrucha, M. "The Complete Toolbox: Handy Things to Do With Your UCAVs." Group 61, MIT Lincoln Laboratory. 15 September 2004.
- [38] Shroer, R. "UAVs: the future. [A century of powered flight: 1903-2003]." *Aerospace and Electronic Systems Magazine, IEEE*. Volume 18, Issue 7. July 2003.
- [39] "Unmanned Aerial Vehicles (UAVs)." *Federation of American Scientists*. <<http://www.fas.org/irp/program/collect/uav.htm>> Accessed 14 January 2004.
- [40] MIT Lincoln Laboratory. Penetrating UAV Study Briefing. 15 November 2003.
- [41] Johnson, R. and O'Neill, M. "Unmanned Aerial Vehicles in Perspective: Effects, Capabilities, and Technologies." United States Air Force Scientific Advisory Board. July 2003.
- [42] Munson, K. "Northrop Grumman BQM-34 Firebee I." Aerial Targets, United States. *Jane's Defence Weekly*. <<http://www4.janes.com>> Accessed 23 February 2005.
- [43] Boeing YQM-94A "Compass Cope B." Unpiloted Vehicles. USAF Museum. <<http://www.wpafb.af.mil/museum/annex/an5.htm>> Accessed 23 February 2005.
- [44] Boeing Condor. Online Museum Exhibits. Hiller Aviation Museum. <<http://www.hiller.org/exhibits/online-exhibits/condor/condor.html>> Accessed 23 February 2005.
- [45] Munson, K. "Northrop Grumman RQ-4A Global Hawk." UAVs, United States. *Jane's Defence Weekly*. <<http://www4.janes.com>> Accessed 15 January 2004.
- [46] Munson, K. "General Atomics MQ-1 and RQ-1 Predator." UAVs, United States. *Jane's Defence Weekly*. <<http://www4.janes.com>> Accessed 15 January 2004.
- [47] Choi, M.G. "Intelligence, Surveillance, and Reconnaissance (ISR) Missions for a Penetrating Unmanned Aerial Vehicle (UAV)." Staff Seminar. MIT Lincoln Laboratory. 7 November 2003.
- [48] "Black Widdow Micro UAV." Defense Update. *International Online Defense Magazine*. <<http://www.defense-update.com>> Accessed 24 February 2005.

- [49] Sheridan, T. B., and Verplank, W. L. "Human and computer control of undersea teleoperators." Technical Report, Man-Machine Systems Laboratory, Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge. Massachusetts, 1978.
- [50] Howitt, S. L. and Richards, D. "The Human Machine Interface for Airborne Control of UAVs." 2nd AIAA "Unmanned Unlimited" Systems, Technologies, and Operations. San Diego, CA. September 2003.
- [51] Cummings, M., and Morales, D. "UAVs as Tactical Wingmen." *Unmanned Systems*. Jan/Feb 2005. Vol 23, No 1. Pages 25-29.
- [52] Novak, L.M. et al. "The Automatic Target-Recognition System in SAIP." *The Lincoln Laboratory Journal*. Vol 10, No 2. 1997. Pages 187-202.
- [53] "ISR Platform Coverage Comparisons." Briefing Slides. MIT Lincoln Laboratory. 7 November 2004.
- [54] Air Force Link. Library, Fact Sheets. <<http://www.af.mil/factsheets/>> Accessed 27 February 2005.
- [55] Zeger, L. "UAV Communications." Briefing Slides. MIT Lincoln Laboratory. 8 March 2005.
- [56] Chant, C. *Air Defense Systems and Weapons*. Brassey's Defence Publishers: New York, 1989.
- [57] "Nonproliferation: Further Improvements Needed in U.S. Efforts to Counter Threats from Man-Portable Air Defense Systems." United States General Accounting Office, Report to Congressional Committees. May 2004.
- [58] Lussier, F.M. et al. *Army Air and Missile Defense: Future Challenges*. RAND: Santa Monica, 2002.
- [59] Hansen, J. *The Development of Soviet Tactical Air Defense*. International Defense Review, Special Series: 14, Air Defense Systems. 1982.

[60] Parsch, Andrea. "Oerlikon/Contraves RSC-51 (MX-1868)." Directory of U.S. Military Rockets and Missiles. Appendix 4: Undesignated Vehicles. <<http://www.designationsystems.net/dusrm/app4/mx-1868.html>> Accessed 2 March 2005.

Appendix A

Surface to Air Missiles

It would be quite helpful if UAVs and other aircraft could fly unhindered into enemy territory. They would be able to obtain valuable imagery, perform strike missions, and return safely to friendly territory. However, this rosy scenario leaves out some of the largest and most sophisticated threats that aircraft face today. Ground-to-air and air-to-air defenses are effective deterrents to penetrating missions. Ground-to-air systems can be as primitive as guns, rockets, or bombs thrown into the air. Anti-aircraft (AA) guns on a tracked or wheeled chassis with large caliber rounds are an improvement. MANPADS, which are portable, short-range SAM systems carried by one or two people, can be even more effective. Components include the launch tube and missile, gripstock, and thermal battery [58]. The U.S. Stinger missile system is one of the most advanced MANPADS, and they represent a definite threat to aircraft. However, the most sophisticated air defenses in existence today are long-range SAMs, especially when used in groups (batteries) and connected to an advanced, long-range radar system.

A.1 Radar Development

A typical SAM system can be described as performing three actions. It detects an incoming aircraft, identifies and tracks it, and then shoots it down if necessary. All three actions incorporate some aspect of radar. Radar works by bouncing electromagnetic waves off of approaching objects and then using the reflected signal to identify the object and determine its range, velocity, and other characteristics. Radar systems can be classified by their function

(surveillance, track, etc...), by their detection range (long, medium, and short), by their frequency band (X, K, etc...), and by a multitude of other categories. This thesis will focus on X-band fire-controlled radar, used frequently in SAM systems.

The basic principles of radar have been in use for millions of years. Bats emit high frequency sounds from their nose, and the reflected echoes are received by the bats' ears. This "ultrasonic sensor" was used by bats to hunt prey and avoid terrain, among other things [33]. A frequent meal for a bat is the tiger moth. However, tiger moths have themselves developed the ability to produce their own sound that interferes with the bats' sound and allows the tiger moth to escape unharmed [33].

The first human foray into radar, which means radio detection and ranging, came in 1887 when a German physicist named Heinrich Hertz began investigations into how radio waves functioned. He found that radio waves could be transmitted through some materials but were reflected by other materials [35]. He was even able to measure the speed of the waves. In 1904, another German named Christian Hulsmeyer developed the Remote Object Viewing Device, which could detect ships that were up to three kilometers away [33]. Many other researchers soon got involved, and in 1930, the U.S. Naval Research Laboratory discovered radar almost accidentally. They were performing measurements on a radio link that stretched across a runway when an aircraft crossed the beam and their radio signal fluctuated sharply [33]. Radar was discovered independently at the same time in the United Kingdom, Germany, the Soviet Union, France, Italy, Japan, and the Netherlands. Throughout the 1930s, radar remained inaccurate, low-powered, and unreliable. However, buoyed by the period of technological innovation initiated by World War II, radar rapidly matured into an accurate and versatile capability by 1945. In the latter half of the 20th century, research intensified, and radar terms such as bistatic, spread spectrum, conical scan, and continuous wave Doppler became a common and integral part of the language. The focus of this thesis is on detection of aircraft via radar, which is explored in more detail in Appendix C.

A.2 SAM Development

At the same time that radar was being developed, rockets and missiles were undergoing a large transformation. They were used in World War I mostly to provide smoke screens or to illuminate the battlefield, but the early 20th century saw the development of the Kettering Bug in 1917, which was one of the first guided missiles. Solid-fueled missile research continued, with the U.S., Great Britain, and Germany leading the way. Liquid-fueled missile tests were performed in the early 1930s, to be followed by an intensification of research during World War II.

Allied bomber attacks on German soil spurred the Germans into developing a number of missiles intended to shoot down Allied aircraft, to include the Enzian, the Feuerlilie, the Shmetterling, and the Wasserfall, to name a few [32]. The Wasserfall used a pressure-fed, liquid-fueled engine. It could fly at Mach 2.5 up to eight miles high and at a range of 17 miles [32]. It was guided and detonated by radio command. Toward the end of World War II, the proximity fuse was created, which uses the Doppler effect to calculate the missile's proximity to its target and then detonate. However, none of these missiles were connected to developing radar systems to allow for more accurate air defenses.

The world's first commercially available and operational SAM was the RSC-50, developed by the Swiss companies Oerlikon-Bührle & Co. and Contraves AG in 1947 and completed in 1950 [60]. More improved versions soon followed, to include the RSC-51, RSC-54, RSC-56, RSC-57, and RSC-58. These missiles were quite successful and used by Switzerland, Sweden, Italy, and Japan. The U.S. performed an initial evaluation of the RSC-51 but did not pursue it further [60]. The RSC-51, shown in Figure A-1 [60], utilized two separate trailers. One trailer contained a tracking radar and a beam transmitter, and the other trailer launched the missile using a liquid-fueled rocket engine [60]. The missile traveled at speeds in excess of Mach 2, could reach altitudes higher than 66,000 ft, had a range of 20 km, and detonated by use of a proximity fuse [60].

The world's second operational SAM was the Russian SA-1 "Guild," which entered service in 1954 [56]. Intended to shoot down high-flying nuclear-armed bombers, it was a great achievement for its time. It was a fixed-site SAM that was deployed in two concentric rings around Moscow. Over 3,200 of these SAMs, shown in Figure A-2, were deployed at one time,

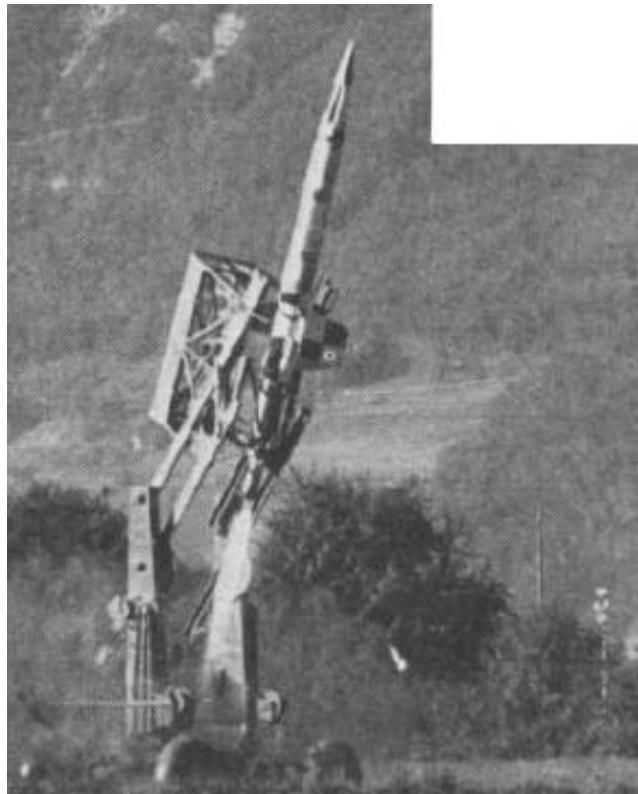


Figure A-1: RSC-51. This SAM was developed by the Swiss companies Oerlikon-Bührle & Co. and Contraves AG in the early 1950s. It utilized two trailers for the radar and the missile launcher.



Figure A-2: SA-1 "Guild." The world's second operational SAM was the Russian SA-1 "Guild," which entered service in 1954. Intended to shoot down high-flying nuclear-armed bombers, it was a great achievement for its time. It was a fixed-site SAM that was deployed in two concentric rings around Moscow.

and a typical SA-1 site contained about 60 missiles in addition to the Yo-Yo radar [34]. The Yo-Yo radar was a fire-controlled radar that could track more than 24 targets at the same time. It was called Yo-Yo because the motion of its six rotating antennas resemble the motion of a toy yo-yo [34].

The Russians soon continued with an aeronautical development of the SA-1 called the SA-2 "Guideline," a long-range, high-altitude, two-stage missile still in use in many countries. It measures over 35 ft in length and travels at speeds in excess of Mach 3.5 [34]. The SA-2 can reach targets as high as 90,000 ft and has a range of approximately 50 km [34]. It is aided by the missile control radar Fan Song and by an early warning radar called Spoon Rest that can detect targets up to 275 km away. The SA-2 (Figure A-3) is significant in the development of SAMs for many reasons. This type of missile (actually, 14 of them) was used to shoot down a U.S. U-2 spy plane flown by Francis Gary Powers on 1 May 1960 near Sverdlovsk, causing a political crisis over his imprisonment and U.S. spy operations in Russia. Ironically, a Russian MiG-19 fighter jet was also shot down in the engagement [34].

The SA-2's role in world affairs was not limited to that one engagement. Over 13,000



Figure A-3: SA-2 Guideline. The SA-2 is a long-range, high-altitude, two-stage missile still in use in many countries. It measures over 35 ft in length and travels at speeds in excess of Mach 3.5. It is famous for having been used to shoot down a U.S. U-2 spy plane flown by Francis Gary Powers in 1960.

SA-2s have been fired in combat, which is more than any other SAM [34]. In 1962, the U.S. discovered nuclear missiles on Cuban soil via their deployment of SA-2 batteries around the sites, which then sparked the Cuban missile crisis. SA-2s have been or are currently used by over 40 nations, to include Russia, China, Vietnam, Cuba, and Egypt. The SA-1 and SA-2 were just the beginning of a long family of Russian SAMs. Other SAM families exist, such as the Chinese HQ series. However, the Russian SAMs represent probably the greatest threat to U.S. interests because of their sophistication and their wide distribution.

The importance of air defenses was highlighted by significant battles around the world. The inability of Russian air defenses to prevent the German Luftwaffe from destroying troop concentrations and key installations from 1941-1945 led to full-scale investments in air defense. North Korean and Chinese military commanders learned this lesson and utilized various types of AA guns along with small-arms fire to defend themselves against attacking U.S. aircraft in the Korean War [59]. The U.S. Air Force eventually lost 544 aircraft to these defenses, which was almost five times the amount of losses attributed to air-to-air engagements [59]. The widespread use of radar equipment in North Korea also gave the defenders an advantage.

Egypt, which had learned from these experiences, had purchased and deployed 18 SA-2 battalions (with six launchers in each battalion) along with a vast array of AA guns [59].

However, on 5 June 1967, the Six Day War began with the destruction of 16 Egyptian airfields and 26 Egyptian radar stations by the Israeli Air Force (IAF) [59]. By the end of the second day, 418 Arab aircraft were destroyed (393 on the ground). This war led to three lessons. The first was the obvious need for effective low-level defenses. IAF planes were able to fly low, under the envelope of the SA-2s. The second lesson was the need for better training since the combat readiness level of the Egyptian SAM operators was extremely low. The third lesson was the demonstrated need for mobile air defense. After the surprise on the first day, the Egyptians had to abandon vast amounts of equipment in their hasty retreat, such as launchers and missiles.

The extensive use of SAMs in Vietnam forced the U.S. to change air tactics. They used F-105s as decoys to get SA-2 sites to radiate in anticipation of a shoot-down. When the SA-2s were detected, the F-105s would shoot anti-radiation missiles at the SA-2 locations, clearing the air for future strike missions [59]. Sometimes, certain areas just had to be avoided due to a high concentration of SAMs. Another tactic that U.S. aircraft used was electronic jamming and other electronic countermeasures (ECM) to SAMs. The North Vietnamese, along with the Russians, learned that these countermeasures could seriously decrease the performance of air defenses, which led to research into frequency diversity and other electronic counter-countermeasures (ECCM) [59]. The shoulder-fired SA-7 "Grail" used heat-seeking technology to destroy low-level attackers, such as helicopters. In return, U.S. helicopters learned to drop flares that would divert the SA-7. Increasing U.S. reliance on helicopters caused their numbers to increase from 300 in 1964 to about 2,400 in 1967 [59]. This caused Russia to put a larger emphasis on low-level defenses to defend ground forces against the slow, low helicopters.

After their humiliating loss in the Six Day War, Egypt had to build up its air defenses once more. The Russians provided a wide variety of weapons, to include numerous AA guns, the SA-7, the SA-3 "Goa," and the SA-6 "Gainful" [59]. The SA-6 is another significant SAM in the Russian SA series. It was a potent weapon when introduced in the 1960s. It is a mobile, armored carrier with a medium-range, low-altitude missile. It uses the Long Track early warning radar, which provides detection capabilities up to 150 km, in conjunction with the Thin Skin height-finding radar [34]. In addition, the Straight Flush radar provides guidance in the initial missile flight stage and continuous-wave illumination of the target to allow the



Figure A-4: SA-6 Gainful. The SA-6 is another significant SAM in the Russian SA series. It was a potent weapon when introduced in the 1960s. It is a mobile, armored carrier with a medium-range, low-altitude missile, and it is used extensively throughout the world.

missile to actively home in on the target. The propulsion of the SA-6 (Figure A-4) is also fascinating. It uses a rocket booster to power the missile to Mach 1.8 and then discards the rocket nozzle while lighting an internal ramjet [34]. This hybrid propulsion system makes the SA-6 extremely responsive and may be why it is used so extensively in many regions throughout the world, to include the Middle East and southern Africa.

By July of 1973, Egypt had built up an integrated air defense (IAD) network that included over 130 SAM sites [59]. Syria and other Arab nations also posed formidable threats. It was clear from the start of the Yom Kippur War on 6 October 1973 that the Arab nations' preparation had paid off. The IAF lost over 100 combat aircraft, mostly due to SAM and AA defenses [59]. However, by the end of the war, Israeli ECM and tactics had improved to the point where the SA-6 had only a meager two-percent probability of kill, or probability that the missile would shoot down the intended target [34]. Another prominent lesson to be taken from this conflict was that the Arab defenses were allowed to remain relatively static. However, Russia and other countries saw this as an anomaly and believed that highly mobile, yet powerful SAMs would be the key to future engagements. Thus, future SAMs in the Russian SA series and other SAMs reflected that belief.

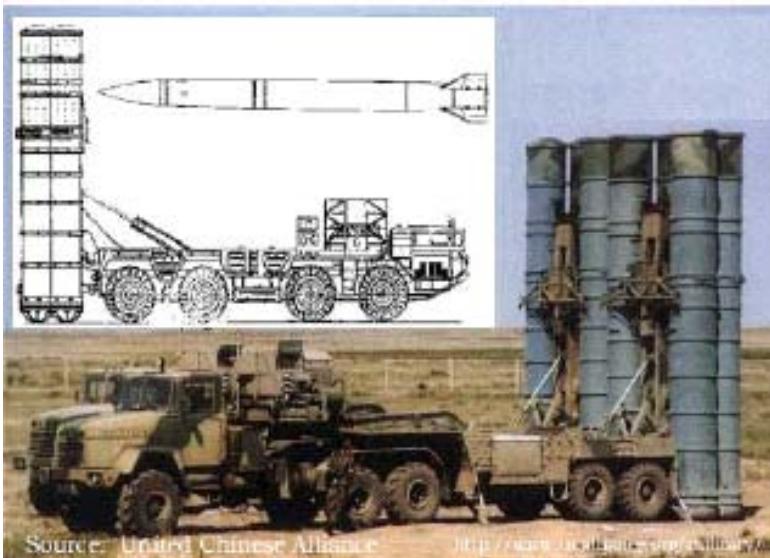
A.3 Modern SAMs

It is clear from the preceding discussion that enemy air defenses, and, to some extent, aircraft tactics, were the result of a long-standing pattern of action and reaction. That pattern continues to this day. The SA-10 and the SA-20 are two extremely advanced SAM systems developed in reaction to perceived capabilities of modern aircraft and weapons. These SAM systems are capable of inflicting serious damage on any penetrating aircraft and should be taken quite seriously.

The SA-10 "Grumble" is the much improved successor to the SA-2 and was introduced in 1980. Its aim is to shoot down low-level, high speed aircraft or missiles. It can be fixed or mobile. The missile uses a single-stage, solid rocket engine that flies at speeds well over Mach 5. It can reach altitudes up to 90,000 ft and can engage targets at a range of over 200 km [34]. The system utilizes the Big Bird radar system to detect low altitude, terrain-masked targets, the Clam Shell radar to acquire targets using extreme ECM, and the Flap Lid to track both the target and the missile and to issue mid-course corrections if necessary [34]. The SA-10, shown in Figure A-5, boasts a single shot kill probability of 0.7 to 0.93 [34]. The SA-10 is a mobile and extremely capable SAM that poses unique challenges to aircraft in the theater, and it has been sold to numerous countries, including China, India, and others.

One of the most advanced SAMs in existence today is the SA-20 "Triumf," scheduled for deployment in 2005. Possibly its most remarkable characteristic is its ability to engage and destroy targets as far as 400 km away, or at twice the range of the SA-10 [34]. This capability has the potential to drastically change the manner in which ISR and strike operations are performed. The SA-20 can fire two different types of missiles. The first missile can reach targets 400 km away and can operate in active or semi-active seeker modes [34]. It was specifically designed to target jammer aircraft or early warning aircraft, such as the AWACS. The associated radar system can reportedly detect targets over 600 km away and engage up to six separate targets at one time [34]. This clearly shows why typical standoff assets will be unable to penetrate very deep into enemy territory. Penetrating UAVs or some alternative will be needed to fly undetected through enemy airspace to avoid the lethal reach of the SA-20 and other SAM systems.

The second missile can engage targets up to 120 km away but has a single shot kill probability



Source: United Chinese Alliance

<http://www.usian-similarity.com>

Figure A-5: SA-10 Grumble. The SA-10 "Grumble" is the much improved successor to the SA-2 and was introduced in 1980. Its aim is to shoot down low-level, high speed aircraft or missiles. It can be fixed or mobile. The missile uses a single-stage, solid rocket engine that flies at speeds well over Mach 5, and it has been sold to numerous countries.

of 0.9 against manned aircraft [34]. Its innovative gas-dynamic control system permits the missile to maneuver at forces over 20 Gs and at altitudes up to 115,000 ft [34]. Human pilots in manned aircraft can withstand 9 Gs at most. Other versions of the missile use an even more advanced guidance and control system, a new high-energy solid fuel motor, and a "transverse control engine," which supposedly enhances the accuracy of the missile in its final stage [34]. Also, the SA-20 missiles are interoperable with the SA-10 and can quadruple the capacity of the launchers. Additionally, the SA-20 (Figure A-6) may be hard to jam since it can "vary the waveform and change polarization of the signal," thus rendering some current U.S. jamming practices obsolete [34]. Finally, the SA-20 missiles are reportedly made from stealth materials, which would give the targeted aircraft less notice and less time to maneuver [34]. SA-20s are currently being marketed to China, Iran, the United Arab Emirates, and other countries.

The evolution of SAM systems throughout the 20th century certainly displays a remarkable trend of technological innovation. Current SAMs, with amazing precision and long reach, have the potential to drastically change the nature of how the air war is fought and won. The



Figure A-6: SA-20 Triumf. Possibly the most remarkable characteristic, among other features, of this SAM is its ability to engage and destroy targets as far as 400 km away, or at twice the range of the SA-10. This capability has the potential to drastically change the manner in which ISR and strike operations are performed.

observed pattern of action and reaction means that today's aircraft must find new ways to avoid detection if ISR and strike missions deep within enemy territory are to be accomplished. Management of an aircraft's radar signature, better operational tactics, and enhanced stealth technology will surely be integral to the success of air warfare in the 21st century.

Appendix B

Unmanned Aerial Vehicles (UAVs)

Unmanned Aerial Vehicles (UAV) have had a large impact on military operations in the past decade, and they now comprise a critical component of the armed forces. Military strategists, commanders, and operators have all championed the capabilities of UAVs and advocated for their continued production, availability, and use. UAVs come in all shapes and sizes with a variety of sensors, weapons, and equipment. They are largely used in ISR operations but have also proven to be effective in hostile, offensive combat missions. Clearly, UAVs offer a formidable and valuable resource for planners of military operations.

B.1 History of UAVs

Many theories exist in regards to what exactly defines a UAV. There remains an equally long list of terms that are synonymous to UAVs, such as drones, Remotely Operated Aircraft (ROA), Remotely Piloted Vehicles (RPVs), and Unmanned Vehicles (UVs), to name a few. For simplicity, the acronym UAV will be used here. The most inclusive definition of a UAV includes powered aerial vehicles which maintain altitude through aerodynamic lift and lack the physical presence of human beings. They can be recoverable or expendable and can be piloted remotely or fly autonomously, meaning without any human aid. Robotic weapons and cruise missiles are even considered to be UAVs. This all-inclusive definition will be used in the following discussion.

B.1.1 Early History

The first development in the history of UAVs occurred in May of 1896 when Samuel Pierpont Langley constructed an unpiloted heavier-than-air vehicle that subsequently flew over the Potomac River [41]. The progress of UAVs continued during World War I with the development of the "Sperry Torpedo," by pilot and inventor Lawrence Sperry. This torpedo began construction in 1916 with the first test flight in March of 1918. Near the end of World War I, two different, semi-autonomous, unmanned aircraft began development. One was a Navy-designed craft called "the flying bomb." The other was developed by the US Army Signal Corps and called the Kettering Aerial Torpedo, or "Bug" [38]. Both of these planes used gyroscopes invented by Lawrence Sperry to control attitude and an aneroid barometer for altitude. The actual airframe for both planes was provided by the Wright Airplane Company [38]. The "Bug" was a small biplane made of wood and fabric with a 180-lb explosive warhead. The plane flew on a preset course and utilized a mechanism that counted engine revolutions to determine when to fold the wings and dive on a target [1]. The plane first flew in October of 1918 at Wright Field but was hampered by developmental problems and delays and was never actually used in combat.

B.1.2 World War II

The United States and Great Britain continued to invest time and effort into UAVs throughout the 1920s, but as Figure B-1 [41] shows, World War II spurred further growth in UAVs. Many nations, including the U.S., Great Britain, Japan, Italy, and Germany, began widespread experimentation with unmanned aircraft during World War II. The people of Great Britain were terrorized beginning in June 1944 by the German V-1 (Vergeltungswaffen-1) cruise missile, nicknamed the "Doodlebug" [1, 41, 38]. Using a barometer for altitude and a magnetic compass for heading, the V-1 utilized an air-driven gyro-based flight control system. The Germans unleashed a barrage of almost 20,000 V-1s against the Allies, who subsequently copied the missile's design, built a similar version, and eventually canceled the program in the 1950s after never using the new missile in combat [38]. Following the German deployment of the V-2 in September of 1944, the United States increased the tempo of its own research endeavors

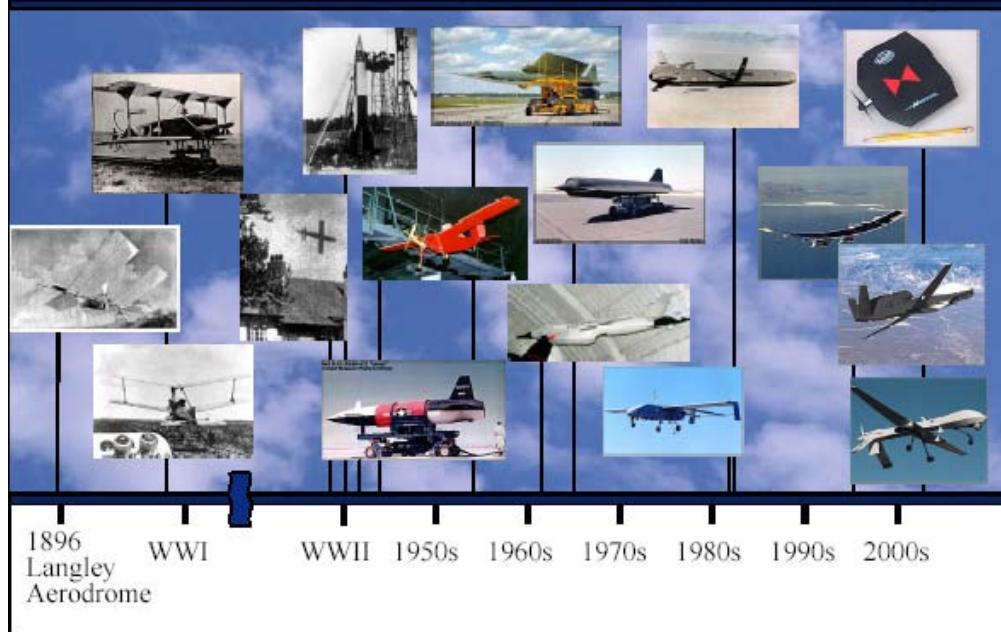


Figure B-1: History of UAV Development. From the flight of the Aerodrome in 1896 to the modern Global Hawk, the development of UAVs has been storied and fascinating.

and developed the Bell B-63 guided missile and the OQ-A Radioplane [41]. These programs unfortunately followed the trend of many systems in that they become operational only at the end of the war, were never produced in mass quantities, and were never deployed in combat. However, the defining capability of UAVs developed during World War II was their expendability, and their use proliferated in situations where manned aircraft would be ineffective or unlikely to return.

B.1.3 Cold War

The development of cruise missiles continued unabated into the 1950s and 1960s with, among others, the F6F-5K Hellcat, a missile launched from the deck of a carrier, the Air Force's XGAM-63, a supersonic missile shot from the air, and the AGM-28 Hound Dog, which could travel over 700 miles [1]. However, an increasing emphasis was placed on recoverable UAVs in the Cold War era. The need to gather intelligence and perform reconnaissance missions became more attractive to military commanders, thus requiring the UAVs to return safely after



Figure B-2: Lockheed D-21 UAV (USAF Museum). The D-21 was a supersonic, air-launched UAV intended to capture images of areas that were unreachable by the SR-71, a fast, high-flying, long-range, stealthy manned reconnaissance aircraft.

performing their mission. A variety of different UAV designs were able to accomplish this objective. One of the first was the Lockheed D-21 "Tagboard" UAV, which was developed and deployed operationally over China in the mid-1960s [1]. The D-21, shown in Figure B-2, was a supersonic, air-launched UAV intended to capture images of areas that were unreachable by the SR-71, a fast, high-flying, long-range, stealthy manned reconnaissance aircraft. However, due to the large success of the SR-71, the D-21 proved to be unnecessary as well as ineffective.

Other UAV systems actually made vital contributions to intelligence operations in the Korean War, the Vietnam War, and other regions. For example, the Northrop-Ventura MQM-57A and MQM-57B were short range, 24-hour, reconnaissance UAVs utilized from 1959 to 1973 by the U.S. Army [1]. However, the Ryan Aeronautics Firebee (BQM-34A and variants) system of UAVs has proven to be the most successful and effective program ever. Developed as a joint Army, Navy, and Air Force project in 1948, the UAV was initially designed to be a target vehicle. Making its first flight in mid-1951, the Firebee (see Figure B-3) is launched by a rocket or in the air and uses a parachute for recovery [1]. To date, over 7,400 Firebees have been built, both in the U.S. and in Japan, and they have been sold to many countries, including Canada, Israel, and Italy [42]. The Firebee has been modified into 20 different variants and has been heavily utilized in strategic and tactical reconnaissance roles. Some variants carried electronic intelligence gathering equipment, and some used television cameras and transmitters



Figure B-3: Ryan BQM-34 Firebee (Northrop Grumman). This is the most successful and effective UAV program ever. Developed as a joint Army, Navy, and Air Force project in 1948, the UAV was initially designed to be a target vehicle and is now used in strategic and tactical reconnaissance roles.

to accomplish their mission. The Firebee also has the distinction of being the first unmanned vehicle to drop ordnance on an enemy site and return to safety. This type of UAV is called an Unmanned Combat Aerial Vehicle, or UCAV. Around 1971, the U.S. Air Force contemplated the use of a UCAV to attack enemy Surface-to-Air Missile (SAM) sites [1], which led to the BQM-34A variant of the Firebee. Later versions included a modular system to switch between reconnaissance and strike roles. The common theme between all the UAVs and UCAVs discussed to this point is that each aircraft was generally launched by a rocket or from the air, flew a pre-determined course, and was recovered by parachute.

B.1.4 Post-Vietnam Development

Although many programs lost funding or were canceled altogether, the late 1970s and 1980s saw widespread continuation of UAV development. The YQM-94A "Compass Cope" was a high altitude, long-range UAV developed by Boeing to compete with a similar system built by Teledyne-Ryan [43]. This UAV was able to take off and land under the direction of a controller at a remote ground station. This design represented a large step toward the more sophisticated UAVs of modern times. Following the Compass Cope was the Boeing Condor, a reconnaissance UAV built in the late 1980s with a gigantic wingspan of over 200 feet (larger than a Boeing 747). The Condor was a stealthy aircraft that could remain in flight for over 80 hours, managing to set an altitude record for a piston-driven aircraft of 67,000 feet [44]. Before its retirement, the Condor was supposedly flown on over 300 secret missions.

Many other UAV systems followed the Condor, but perhaps the most important ones are the RQ-4A Global Hawk and the RQ-1A Predator. The Global Hawk, built by Northrop Grumman, is a high-altitude (50,000-60,000 feet), long-range (15,535 miles maximum), and long-endurance (36 hours) UAV with a 116-foot wingspan, an allowable payload of 2,000 lbs, and a cruise speed of 343 knots [45]. The Global Hawk, shown in Figure B-4, made its first flight on 28 February 1998 at Edwards Air Force Base in California. By May of 2001, Global Hawk had accrued over 330 hours in FAA-controlled civil airspace and on 15 August 2003 became the first UAV to be granted a national Certificate of Authorization to fly routinely in U.S. national airspace [45]. The Global Hawk is intended for long-range missions that require wide-area surveillance. To support that objective, the aircraft is equipped with Electro-Optical (EO),



Figure B-4: RQ-4A Global Hawk (C.W. Controls, Inc.). The Global Hawk, built by Northrop Grumman, is a high-altitude (50,000-60,000 feet), long-range (15,535 miles max), and long-endurance (36 hours) UAV with a 116-foot wingspan, an allowable payload of 2,000 lbs, and a cruise speed of 343 knots. It has been used very effectively in recent military operations.



Figure B-5: GA RQ-1A Predator (Edwards AFB). The Predator is a short-range (2,302 miles), long-endurance (40 hours), and low-flying UAV with a loiter speed of 117 knots. It is unique in being the only operational UCAV within the US military.

InfraRed (IR), and Synthetic Aperture Radar (SAR) sensors. The Global Hawk has been used quite effectively in recent military operations, including providing U.S. Central Command with over 15,000 images over about 1,000 hours of flight time during Operation Enduring Freedom in Afghanistan [45].

The General Atomics RQ-1A Predator grew out of an earlier General Atomics UAV called the Gnat. In contrast to the Global Hawk, the Predator is a short-range (2,302 miles), long-endurance (40 hours), and low-flying UAV with a loiter speed of 117 knots [46]. Making its first flight on 3 July 1994, the Predator has been the workhorse of the UAV reconnaissance community, logging over 50,000 hours by October of 2002. It has been used in Bosnia, Afghanistan, and Iraq [46]. The Predator (see Figure B-5) is also unique in that it (or a variant) is the only operational UCAV within the U.S. military. On 3 November 2002, a Predator used its attached Hellfire missiles to kill suspected terrorists in a civilian vehicle about 100 miles east of Sana'a, the capital of Yemen [46].

B.2 UAVs Today

In 100 years of UAV development, over 1,500 different UAVs have been created. That is a staggering number, considering the small fraction that appeared in the previous brief history. The advances in technology that have occurred since May of 1896 have allowed modern UAVs to be smarter, faster, more persistent, more adaptable, and simply better at accomplishing their mission. This is not to say that UAVs are perfect and have reached the peak of their development. Nothing could be further from the truth. Aside from combat operations, simple flight control issues and human-machine interface problems account for the majority of UAV losses [1, 41]. However, the plethora of UAVs in existence today warrant some degree of admiration, if only for their infinite variety.

B.2.1 Types of UAVs

Due to the diversity in UAV designs, it remains difficult to classify them, and numerous taxonomies exist with regard to UAVs. One common method is to differentiate between UAVs based on their altitude and velocity. This system places UAVs into groups that also happen to share other attributes, such as size, payload capacity, and endurance, although this is not always the case.

The first type of UAVs travel with large velocities at high altitudes. An example is the previously mentioned Lockheed D-21, which could attain speeds of around Mach 3 at altitudes of 90,000 feet or greater. UAVs in this group tend to hold medium-sized payloads and achieve medium levels of endurance. For example, the D-21 has a payload capacity of 190 kg and an endurance of 1.6 hrs. The next class of UAVs travel at smaller velocities and at lower altitudes. A common example is the Global Hawk with a cruise speed of 343 knots at altitudes around 60,000 ft. This group of UAVs has a larger payload capacity (900 kg for Global Hawk) and longer endurance (36 hrs) than the first group. Figure B-6 [47] illustrates all the different types of UAVs.

The next group of UAVs, like the Predator, fly at a lower altitude (25,000 ft) with a smaller velocity (117 knots). This group also follows the trend for endurance with the Predator being able to loiter for over 40 hours, greater than both the Global Hawk and D-21. However, with

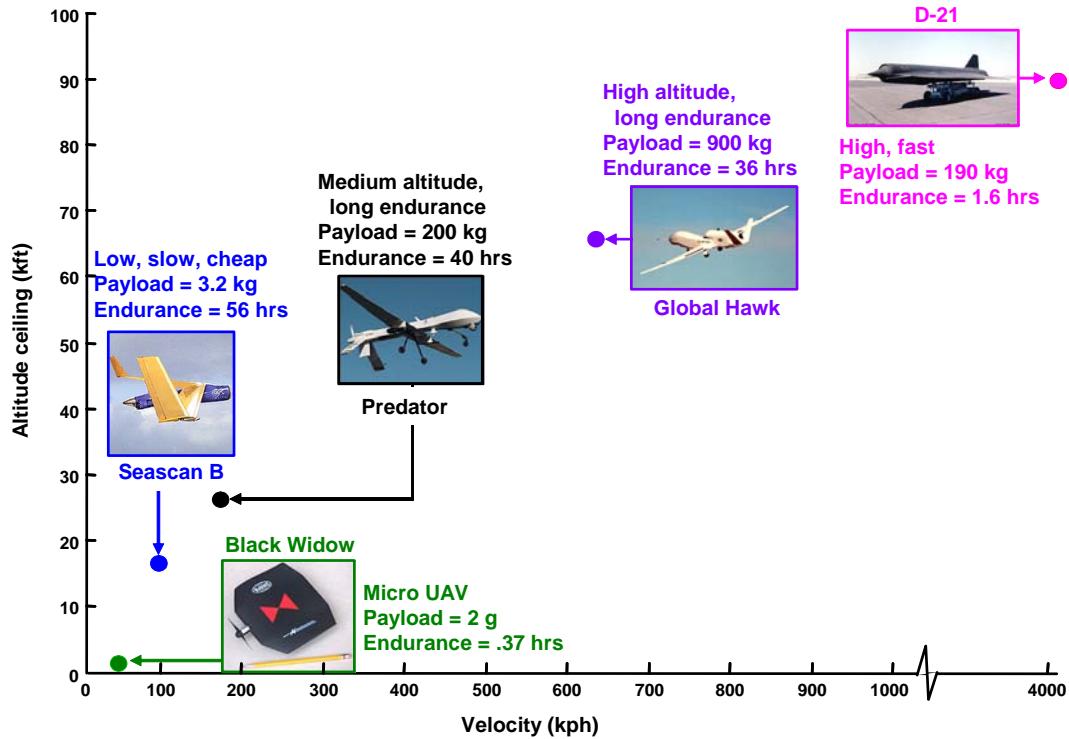


Figure B-6: Classes of UAVs. Micro UAVs such as the Black Widow have a low altitude and low velocity. Other types of UAVs exist all the way until the high, fast UAVs appear, such as the D-21. This variety of UAVs allows many options for military commanders.

regard to payload, the Predator can only carry 200 kg. This is the start of a trend toward smaller payloads, which makes sense since the actual size of the UAVs in each group decreases along with their ability to carry heavier items. The next group flies at altitudes lower than 20,000 ft and at velocities less than 100 knots. An example is the Seascan, a sea reconnaissance UAV built by the Insitu Group to provide aid to services such as search and rescue and coastal patrol. This UAV has a small payload of only 3.2 kg but an amazing endurance of 56 hours [47].

The last group of UAVs are quite promising in what they can offer and are an area of extremely active research. This group is termed Mini- or Micro-UAVs. Some measure as long as two feet across and weigh more than 10 lbs, but others take advantage of microelectronics to reduce the size and signature of the UAV. For example, the AeroVironment Black Widow, shown in Figure B-6, can only carry a tiny payload of 2 grams and has an endurance of only 22 minutes [47]. In addition, it can fly to an altitude of 769 ft at a maximum velocity of about 40 knots. The total weight of the Black Widow is 50 grams, with 25 of those grams being the weight of the primary batteries [48]. Despite its small size, or perhaps because of its small size, the Black Widow and many other Micro-UAVs are an increasingly helpful source of information to aid military commanders.

The general trends within these UAV groups are clear. Looking at Figure B-6, the higher and further to the right one is on the chart, the larger, higher, and faster the UAV tends to be. Inversely, the UAVs that are lower and further to the left on the chart are smaller, slower, and lower. In addition, the endurance of the UAV increases as one moves to the left while the payload capacity decreases. A final trend of interest that one may expect is that the cost per UAV increases as well, going from \$1,000 or less in the left hand corner to millions of dollars in the upper right hand corner. The groups of UAVs presented here are not complete, nor are they set in stone. Many UAVs, because of their diversity, fall in gray areas and cannot be classified easily. However, the taxonomy presented here enables a simple and better understanding of the wide variety of existing UAVs.

B.2.2 Control Methods

One particular area of active research and debate is how UAVs are controlled. This area can be split into three separate categories: remote, semi-autonomous, and autonomous [37]. Each of these methods have specific advantages and disadvantages. Remote control implies direct and immediate control over the movements and actions of a UAV through a continuous datalink from a controller located in a ground station. One important aspect of remote control is the problem of latency, which describes the time delay between the ground controller's commanded action and the physical occurrence of the action with respect to the UAV. The Predator has a three-second lag time when in the direct line-of-sight of the ground station, and that time increases as the distance between the controller and the UAV increases [37]. That delay may not appear to matter, but in hostile, combat situations, every second is needed to avoid being shot down by enemy SAM sites or other threats. However, remote control is quite adaptive in that the UAV controller can respond quickly to a changing environment. The downside is the latency problem, the high taskload imposed on the ground controller, and as well as the need for a continuous communication link between the controller and UAV, which is subject to failure and enemy interception.

Semi-autonomous control allows the UAV to maintain its own flight controls, avoid terrain, and perform various other low-level functions [37]. Commands are then sent via the ground station whenever necessary to alter the flight path, take images, or accomplish a different task. Semi-autonomous control is especially appropriate when the ground controller's taskload is high because the autonomy given to the UAV allows the controller to focus on other tasks and accomplish much more. Also, communication with the UAV can be less continuous and more discrete since the datalink is only needed when instructions need to be sent. Autonomous control essentially puts the entire flight under the control of the UAV. The prime example of this type of control is a cruise missile [37]. With autonomous control, the mission is pre-planned and non-adaptive, which leaves no room to maneuver in a changing environment. However, for certain missions with stable conditions, an automated UAV might be the best option.

UAVs don't necessarily have to utilize strictly one method. The combination of two or more methods may prove to be more effective than simply relying on one. For example, the Predator is remotely controlled, but if that continuous datalink fails, it is programmed to autonomously

Automation Level	Automation Description
1	The computer offers no assistance: human must take all decision and actions.
2	The computer offers a complete set of decision/action alternatives, or
3	narrowes the selection down to a few, or
4	suggests one alternative, and
5	executes that suggestion if the human approves, or
6	allows the human a restricted time to veto before automatic execution, or
7	executes automatically, then necessarily informs humans, and
8	informs the human only if asked, or
9	informs the human only if it, the computer, decides to.
10	The computer decides everything and acts autonomously, ignoring the human.

Figure B-7: Ten Levels of Automation. Sheridan and others divided machine automation into 10 different levels. At the bottom (level 10) is where the computer makes all decisions. The top (level 1) is where the human completes all tasks, and the computer offers no assistance. Many tasks fall somewhere in between those two extremes.

return to a certain point in order to regain communication.

The methods described here actually have a lot more to do with the interaction between humans and machines than with the control of UAVs. In fact, the terms used can be directly related to terms used frequently within the human factors community, which come from a division of automation into ten different levels, shown in Figure B-7 [49]. An automation level of 10 means that the computer (or UAV) decides everything and ignores the human (ground controller), which corresponds to autonomous control. Level 5 means that the computer executes an action with the approval of the human, corresponding to semi-autonomous control, and level 1 signifies that the computer does nothing and allows the human to make all the decisions and take all the actions, which implies remote control.

In addition to the control method, a three-tiered pyramid scheme can be used to analyze the level of interaction between the UAV and the controller (Figure B-8). At the bottom of the pyramid are tasks that can be automated within the UAV, such as following waypoints or maintaining position in a formation. The second tier consists of more complex, knowledge-based decisions, such as a search and destroy directive, and the third tier consists of strategic decisions

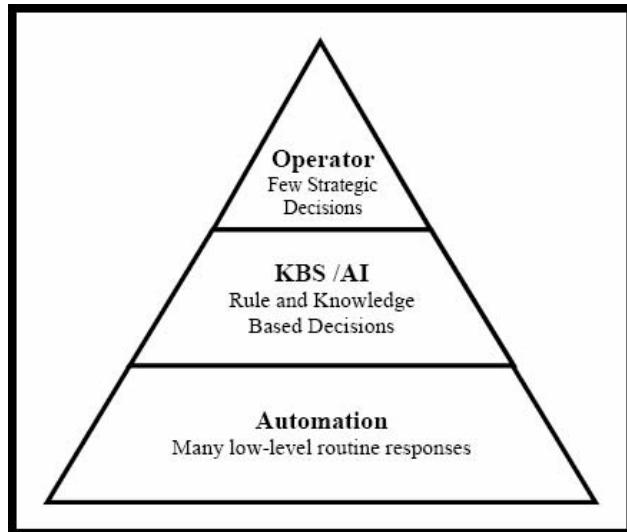


Figure B-8: Levels of Interaction. This pyramid represents different amounts of interaction between a machine and a human. The bottom represents low-level functions that can be automated within a machine. The middle level represents more knowledge-based decisions, and the top level represents the few strategic decisions that require the most interaction.

made by the controller, such as directing the UAV to attack a certain target or approving a target the UAV has selected. Recent studies have suggested that a management-by-consent level of automation (semi-autonomous control) combined with a medium level of interaction may allow for the most effective and efficient control of UAVs in hostile, combat situations [51].

B.2.3 Why UAVs?

One simple question the reader might be asking is why UAVs are so important as compared to other similar manned platforms. Simply put, UAVs possess very unique advantages over manned aircraft. They are often smaller because they are not constricted by the larger size needed to accommodate a pilot. Many design limitations on airframe construction are then removed because the UAV has no need for a pilot, life support system, canopy, flight controls, instruments, displays, and many other items [1]. Eliminating that equipment implies a smaller aircraft, such as the Predator or the Firebee, which allows the UAV to be optimized for lower observability and leads to a reduced radar signature. Conversely, that extra space can also be used to field more sensors and equipment and enable the UAV to be more effective than a

manned aircraft.

The absence of a human pilot allows UAVs to be much more maneuverable and less constrained by human physiological limits. Without injury, human beings can on average withstand only 9 Gs of force, where one G (short for gravity) is equal to the weight of one atmosphere. UAVs can be designed to withstand more than 30 Gs without damage to the aircraft, which makes them much more applicable to missions where sustained forces of 9 Gs or more are necessary. However, current advances in missile technology have now made this advantage irrelevant [1]. The only combat situations that might require high-G maneuvering are within-visual-range "dogfights" between two aircraft and missile avoidance. The vast reach of current air-to-air missiles eliminates the necessity for dogfighting, and the maneuvering and seeking abilities of these missiles make last ditch, high-G, missile avoidance maneuvers futile [1].

UAVs can also operate and maintain a presence for much longer periods of time. The typical endurance of a UAV, as previously illustrated, is over 36 hours and can even exceed 50 hours, which is far beyond the capabilities of manned aircraft. Additionally, a UAV can operate in environments where humans cannot, such as radioactive or contaminated areas. Human flight at altitudes higher than 50,000 feet require pressure suits and special procedures. UAVs would think nothing of flying at those altitudes. UAV accidents and mishaps also cause less politically charged situations. When Francis Gary Powers' U-2 was shot down over Russia on 1 May 1960, a political crisis ensued over U.S. attempts to bring him home, and the nation learned the price it had to pay for the capture of a U.S. pilot in enemy territory [1]. The crash or shoot-down of a UAV, however, causes little political upheaval or emotional outbursts and simply adds to the price tag of the current military operation. A final advantage the UAV has over manned aircraft is the price tag. While not necessarily less expensive in all cases, UAVs have the potential to greatly reduce the cost per aircraft if significant improvements can be made to the current acquisition and operational phases of UAV development [41].

B.2.4 UAV Roles

UAVs and UCAVs are well suited for a variety of missions, to include ISR missions, tactical reconnaissance, deep strike, escort, suppression of enemy air defenses (SEAD), and many others. UAVs have proven their worth in a variety of ISR missions, to include battlefield surveillance

and long endurance tracking. These are mostly uneventful missions that humans in manned aircraft would be happy to delegate to a UAV. Tactical reconnaissance goes one step further and increases the risk to the UAV. This would include missions such as pre- and post-strike reconnaissance, battle damage assessment (BDA), or covert deep reconnaissance [1]. Any air defenses in the areas under surveillance would be on alert, thus turning a relatively safe mission into a dangerous one [1]. The low observability of UAVs and UCAVs lends itself quite well to this mission.

Deep strike involves delivering ordnance against either fixed or mobile targets. UCAVs could destroy their assigned target, take BDA themselves, and return to safety. Alternately, they could be instructed to strike another target. These missions are also dangerous, but using a UCAV eliminates the risk for loss of human life, and real-time imagery could be provided [37]. Another possible UCAV mission is as an armed escort. A UCAV or group of UCAVs could augment the weapons capacity of a flight of manned aircraft, allowing the entire group to be more effective and lethal. The manned aircraft could also return to other, more important tasks and let the UCAV drop the ordnance itself after the assignment of targets.

SEAD is the most prominent of roles for which UCAVs have been considered. To gain air superiority in a region, enemy air defenses must be suppressed or destroyed first. This is neither simple nor easy, and some have described this mission as a duel in which the air defenses usually have the upper hand [1]. However, if the UCAV can operate and react quickly and take into account multiple variables as rapidly as human beings can, then it can provide an effective and safe (from the human perspective) solution to SEAD and allow manned aircraft to patrol the area unhindered by SAMs and other threats. These are just a few of the multitude of missions for which UAVs and UCAVs may be suited.

B.2.5 Current Operations and Future Growth

An analysis of current military operations supports the increasing role of UAVs in the military. Operation Iraqi Freedom (OIF) recently demonstrated the widest use of UAVs in any operation ever. Over ten different UAVs were employed, to include the Global Hawk (1), Predator (17), Hunter (16), Shadow (9), Pioneer (20), FPASS (28), Dragon Eye (20), Silver Fox (6), and Pointer (25) [36, 41]. Additionally, several different Micro-UAVs were utilized by ground forces

UAV Type	Sorties Flown	Hours Flown	Targets Located	Additional Targeting Assistance
Global Hawk	12	280	-13 SAM batteries -50 launchers -300 canisters -300 tanks (38% of known armor) -70 missile transporters	-3700 images taken
Predator	93	1354	-6 targets directly serviced	-Buddy lasing -Talk-on targets for other strike assets
Hunter	151	762	-Artillery -Tanks -Rocket launchers (BM-21, Astros)	-Cross Service cueing for high value target destruction (USMC/USAF)
Shadow 200	150	597	-Technical (weaponized trucks) -Tanks -Armored personnel carriers -Paramilitary personnel	-Conducted force protection, route reconnaissance, security/support prior to raid missions as well as normal RSTA operations
Pioneer	-	1286	-Artillery/Tanks -Rocket launchers -Personnel	-POW rescue, close air support, damage assessment

Figure B-9: UAVs in Operation Iraqi Freedom (OIF). UAVs such as the Global Hawk, the Predator, the Hunter, and many others performed many tasks in OIF such as taking images, identifying targets, close air support, and many other functions. These operations involved many sorties over an extended period of time.

during the engagement [41]. Together, these UAVs provided intelligence preparation of the battlespace, persistent surveillance, broad area search, target identification and designation for weapons employment, strike, battle damage assessment, force protection, and some classified missions as well [36]. Figure B-9 [36] provides more details about the performance and services provided by UAVs in OIF. The result of these UAV efforts was a persistent availability of ISR data, which is crucial to military commanders in rapidly changing environments. Although much improvement is needed, the integration between manned and unmanned systems proceeded relatively smoothly, and UAV losses were much lower than expected (3 Predators, 2 Hunters, and 1 Pioneer) [36].

Looking at the future provides a window into the intensification of efforts to develop and improve more UAV systems. Analysts predict that there will be 16 fully operational UAV

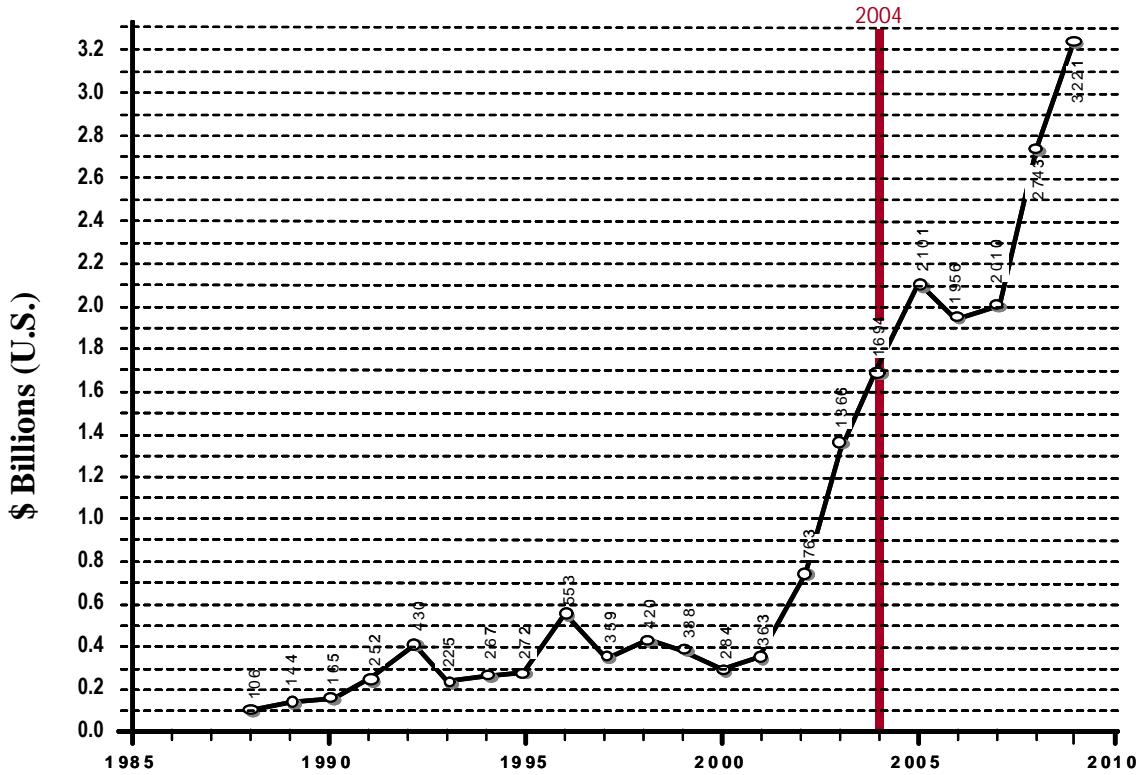


Figure B-10: DoD UAV/UCAV Funding. This graph illustrates the exponential increase in funding for UAVs and UCAVs by the Department of Defense (DoD), clearly revealing its burgeoning interest in UAVs.

and UCAV programs within the next 10 years [36]. Figure B-10 [36] illustrates the exponential increase in funding for UAVs and UCAVs by the Department of Defense (DoD), clearly revealing its burgeoning interest in UAVs. It is clear that UAVs and the technology that supports them are not perfect: nor are they ready to replace manned aircraft. However, it is equally obvious that they render many unique capabilities available to military commanders. Increasing effort must be devoted to the study and development of UAVs so that we do not find ourselves, in the words of one airman, in "an unenviable position: instead of employing these weapons, we will find ourselves facing them" [1].

Appendix C

Radar Systems

Radar can be classified and organized in a variety of ways, and it is used in numerous applications. Remote sensing of the environment involves weather observation, planetary observation, below-ground probing, and mapping of sea ice [31]. Air traffic controllers rely heavily on radar to properly route aircraft into and out of airports. Of course, most drivers have encountered a form of radar that is used by police officers to enforce speed limits. Pitchers in baseball use radar to determine the speed of each pitch. Space vehicles use radar for rendezvous and docking. In addition, larger radar systems are able to track satellites and other space objects. There exist many more fields to which radar has been applied, but the vast majority of radar use and development is attributable to the military. Radar is vital to military air defense and offensive weapons. Different radar systems perform surveillance, fire-control, guidance, and a variety of other purposes. Some aspects of intelligence gathering rely on radar, and most aircraft use some form of terrain-following radar to avoid collisions, so it is clear that radar has a large imprint in today's world.

C.1 Characteristics of Radar

There are many positive aspects of radar. It can detect objects at long range as well as perform tracking. Unlike EO sensors, radar is not affected by clouds or adverse weather. It can detect moving targets, take images of fixed targets, and rapidly scan over wide areas with electronic scanning antennas. It can operate at large wavelengths, and it propagates through

the atmosphere incurring relatively small losses as compared to acoustics [30].

However, there are some detractions to using radar. Large, heavy antennas are necessary, as are high-powered transmitters in some cases. Radar systems typically consist of complicated electronics that use a lot of power and cost a large amount of money. Microwaves also do not propagate very well around obstacles or through water, trees, or foliage [30]. In addition, radar's task of identifying and tracking an unknown target is made much more challenging due to the countermeasures a target can take. Many aircraft know that they're being detected, can locate the radar site, and can use ECM along with chaff, flares, and anti-radiation missiles to render the radar obsolete. Additionally, an aircraft can significantly reduce its Radar Cross Section (RCS), or the degree to which the target reflects the microwaves, by using stealthy materials and enhancing the aircraft design.

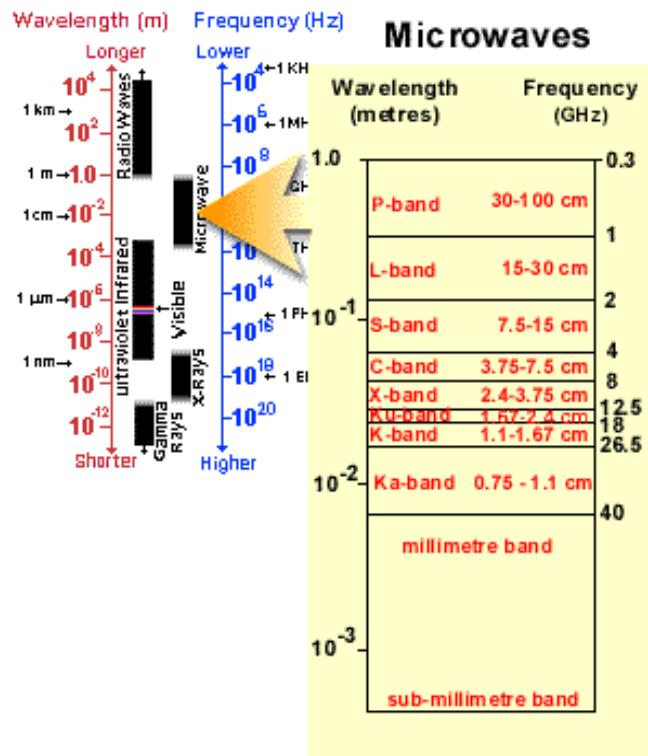
C.1.1 Radar Wave Basics

Radar operates in the microwave region of the electromagnetic spectrum, between 7.5 mm - 100 cm in wavelength and between 3 MHz - 300 GHz in frequency. This region, shown in Figure C-1 [29], is located directly between infrared waves and radio waves. As can be seen in Figure C-1, each segment of the microwave region is divided into separate bands, which are each denoted by a capital letter. These letters may be confusing, especially in comparison with other spectrum charts, since several different letter-band designations are in use. However, the majority of these bands were created by the military to maintain secrecy during World War II, and radar engineers continued to use these band designations for convenience after the war [31].

Each radar wave propagates at the speed of light, c , which is equal to approximately 3×10^8 m/s. The wavelength and frequency of an electromagnetic wave can be related through Equation C.1, where f is frequency and λ is wavelength.

$$\lambda = \frac{c}{f}. \quad (\text{C.1})$$

The most common radar waveform is a series of rectangular-shaped, pulse-modulated sine waves, often called a pulse train [31]. Care must be taken to allow sufficient time between the



© CCRS / CCT

Figure C-1: Electromagnetic Spectrum. The spectrum is divided into many groups, such as radio waves, microwaves, visible, or gamma rays. Microwaves are further divided into different bands relating to their wavelength or frequency. These bands were initially created by the military for use in wartime but have continued to be used. Fire-control radar for many SAM systems utilize X-band radar, between 8 and 12 GHz in frequency.

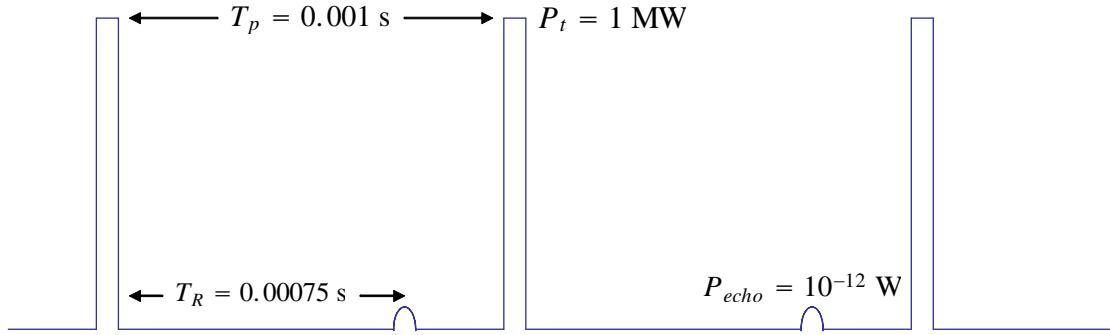


Figure C-2: Pulse-Modulated Waveform Example. This is an example of a typical waveform for a medium range air surveillance radar. In this example, the time between the transmissions T_p is one millisecond, the peak power P_t is one megawatt, the target's echo is 10^{-12} watts, and T_R is 0.00075 s.

transmission of each signal. The reflected waveform, or echo, must return between the two transmissions, or else it could be mistakenly associated with the next transmission, resulting in an inaccurate range measurement. That range is calculated by recording the amount of time, T_R , that it takes for the signal to travel to a target and back. Using that value in Equation C.2 along with c , the range can then be calculated.

$$R = \frac{cT_R}{2}. \quad (\text{C.2})$$

An example of a typical waveform for a medium range air surveillance radar is shown in Figure C-2 [30]. In this example, the time between the transmissions T_p is one millisecond, the peak power P_t is one megawatt, the target's echo is 10^{-12} watts, and T_R is 0.00075 s. Ignoring other factors, the target would be 112.5 km away, and the radar would have a maximum detection range of 150 km in this simplified situation. Both quantities are found by plugging the respective times into Equation C.2.

The output of the receiver might look something like Figure C-3 [30] after a period of time. The threshold level determines valid detections and is set somewhere above the average value of the noise, which is ever-present throughout the electromagnetic spectrum. Points A and B in Figure C-3 are valid detections of a target whereas point C is a missed detection. The threshold level allows tradeoffs between missed detections and false alarms.

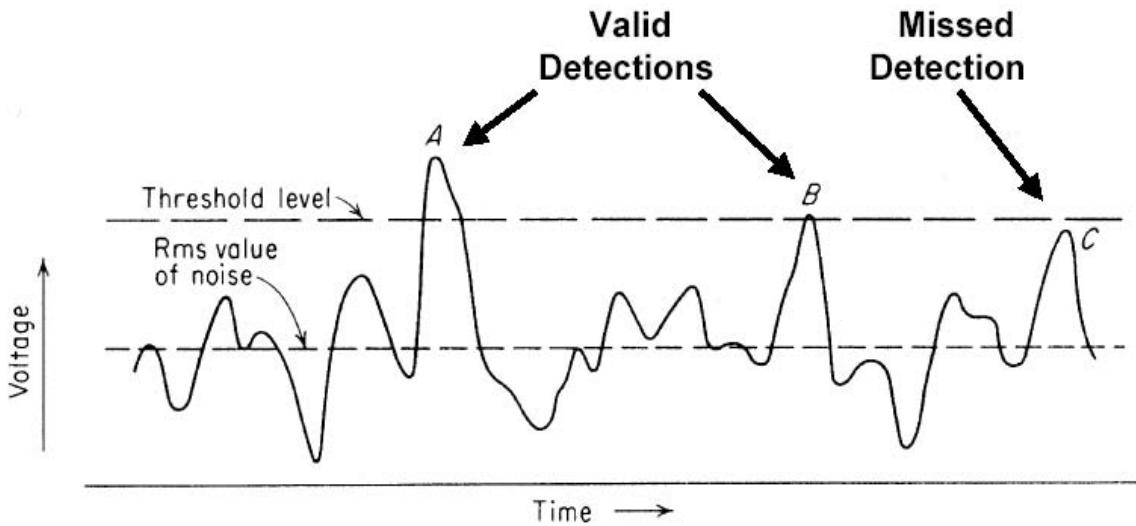


Figure C-3: Envelope of Radar Receiver Output vs Time. This is an example of what a radar receiver might look like. Points A and B, which lie above the threshold level, are valid detections whereas point C is a missed detection because it is well above the noise but still below the threshold that was set.

C.1.2 The Doppler Shift

Another important characteristic of radar is its ability to measure the Doppler frequency of a target. A simple example to illustrate this ability would be a passing ambulance. A person standing on the side of the street would hear the pitch of the ambulance's siren increase as it gets closer and closer. As the siren passes the person and gets farther away, the person would hear the pitch of the siren decrease in a similar fashion. This change in pitch is the result of a change in the frequency of the sound waves being received by the ears of the pedestrian. As the ambulance gets closer, the intervals between each sound wave become smaller, and as the ambulance recedes, the intervals between the waves increase. An Austrian mathematician and physicist by the name of Christian Doppler discovered this anomaly in 1842, and the Doppler shift, as it became known, had a profound effect on many technologies, including radar systems, astronomy, medicine, and speed enforcement [28].

The principles of the Doppler shift hold true for electromagnetic waves as well as sound waves [31]. If the distance between the radar location and the target is R , then the total

number of wavelengths in the round-trip path between the radar and target is

$$\frac{2R}{\lambda}. \quad (\text{C.3})$$

Since each wavelength implies a phase change of 2π radians, then the total phase-change, ϕ , in the round-trip path is

$$\phi = 2\pi \times \frac{2R}{\lambda} = \frac{4\pi R}{\lambda}. \quad (\text{C.4})$$

The focus of this thesis is on radar associated with SAM sites. Thus, the radar will be looking at a moving target. This means that R is changing along with the phase. Taking the derivative of Equation C.4 with respect to time results in a formula for the rate of change of phase, also known as the angular frequency, ω_d .

$$\omega_d = \frac{d\phi}{dt} = \frac{4\pi}{\lambda} \cdot \frac{dR}{dt} = \frac{4\pi v_r}{\lambda}. \quad (\text{C.5})$$

In Equation C.5, v_r is equal to $v \cos \theta$, where θ is the angle between the radar line-of-sight and the target's velocity vector. Also, since $\omega_d = 2\pi f_d$, the Doppler frequency f_d can be found as follows:

$$f_d = \frac{2v_r}{\lambda}. \quad (\text{C.6})$$

A positive Doppler frequency means the target is approaching whereas a negative Doppler frequency means the target is receding. A plot of Doppler frequency vs target radial velocity for different frequency bands is shown in Figure C-4 [30].

C.1.3 Radar Block Diagram

There exists many ways in which the process of radar can be analyzed. It can be viewed with respect to hardware, software, or interfaces. However, the simplest way to depict the radar process is by listing all of the most important actions or components and then linking them all together in a diagram. That is shown in Figure C-5 [30]. A waveform is generated by the ground radar and then transmitted through the antenna to the target in the air. The target has a certain RCS, which causes it to reflect the radar waves back to the antenna. The receiver picks up the target echo, translates the analog signal into a digital one, and sends that

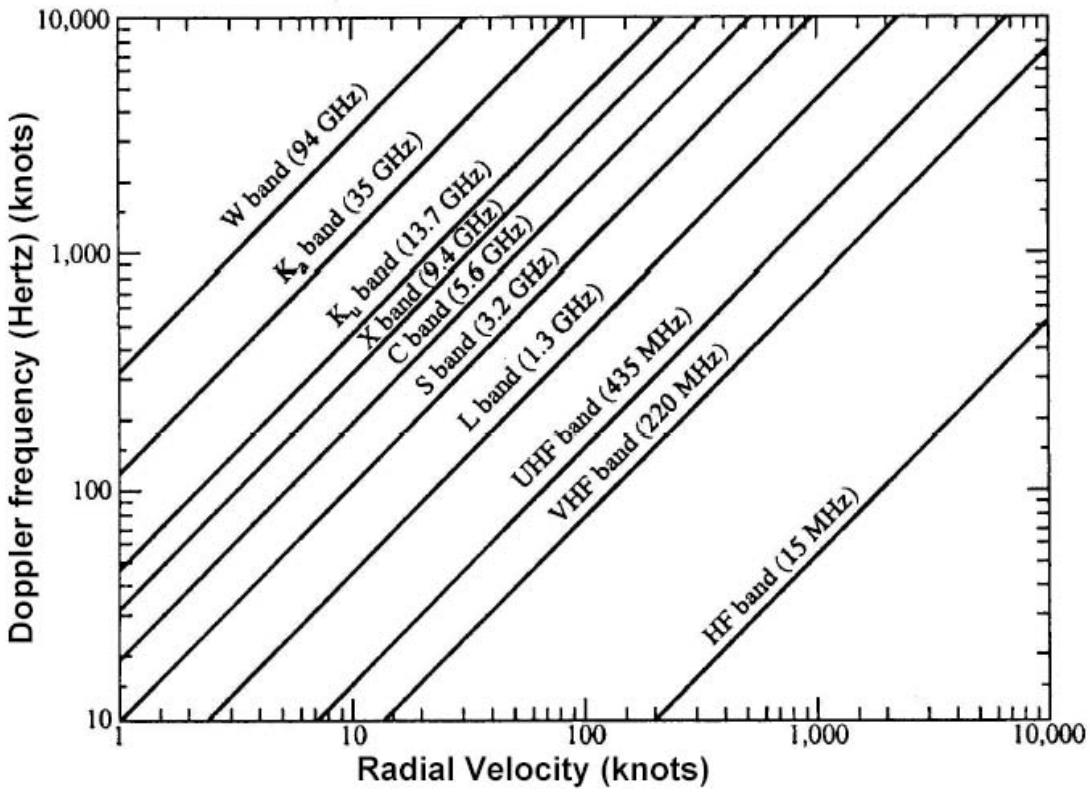


Figure C-4: Plot of Doppler Frequency vs Radial Velocity. The left axis shows Doppler frequency in Hz, and the bottom axis shows the radial velocity of a target in knots. Plotted on the graph are all the various frequency bands and their associated frequencies. Thus, using this graph, one could determine the radial velocity of an approaching aircraft if the radar frequency is known.

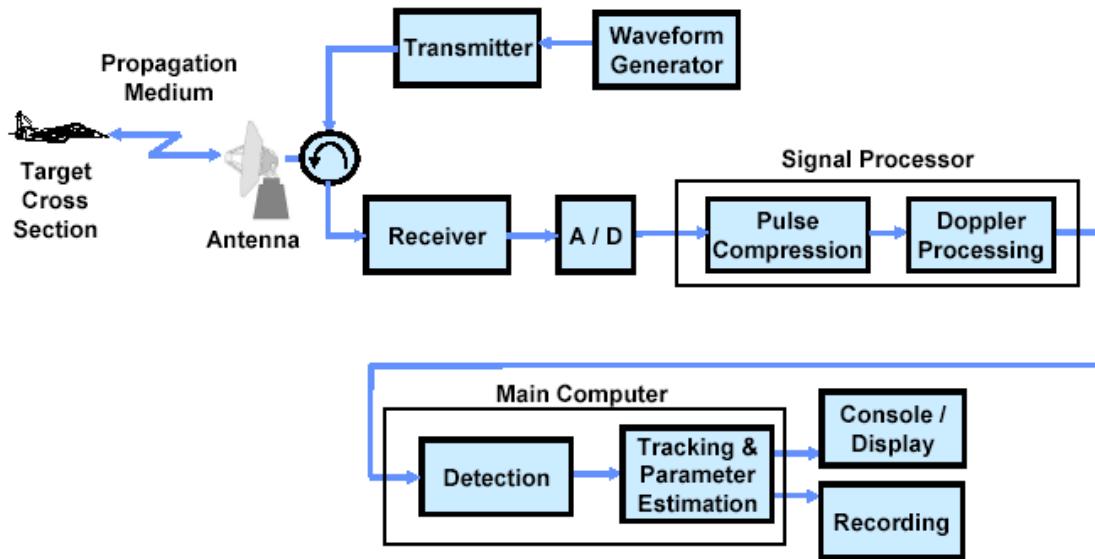


Figure C-5: Block Diagram of Radar. A waveform is generated by the ground radar and then transmitted through the antenna to the target in the air. The target has a certain RCS, which causes it to reflect the radar waves back to the antenna. The receiver picks up the target echo, translates the analog signal into a digital one, and sends that digital echo to the signal processor, which compresses the pulse and finds the Doppler frequency. The main computer then receives the processed signal and determines whether a detection has been made. It will perform tracking and parameter estimation, record the results, and send them to the display, where the human operator makes decisions based on the processed target information.

digital echo to the signal processor, which compresses the pulse and finds the Doppler frequency. The main computer then receives the processed signal and determines whether a detection has been made. It will perform tracking and parameter estimation, record the results, and send them to the display, where the human operator makes decisions based on the processed target information.

C.2 Radar Cross Section

The Radar Cross Section (RCS), σ , of a target is the magnitude of the echo signal reflected by the target and received by the radar [31]. It has also been defined more formally as "the area intercepting that amount of power which, if radiated isotropically, produces the same received power in the radar" [30]. RCS is an area, and its units are typically in meters squared. It is also common to see RCS in terms of decibels, where $RCS_{dBsm} = 10 \log_{10} RCS_{m^2}$. Accurate knowledge of target RCS is vital not only to those operating the air defense systems but also to those targets who are actively seeking to avoid detection.

Typical average RCS values for aircraft range anywhere from -70 dBsm to 25 dBsm ($10^{-7} m^2$ to $300 m^2$). Figure C-6 [30] shows comparative RCS values for birds all the way up to aircraft carriers. The values for some aircraft can get even lower given the advent of numerous stealth technologies. However, Figure C-6 does give a good comparison between objects of different size and shape.

C.2.1 Simple Targets

Objects such as cylinders or cones are relatively simple targets for which analytical expressions exist to calculate the RCS. In particular, a sphere is an excellent example because it radiates isotropically in all directions since it has the same exact shape no matter what aspect angle is presented. The maximum RCS for a sphere is πa^2 , where a is the radius [31]. Figure C-7 [30] shows the normalized RCS of a sphere as the circumference (measured in wavelengths) is increased. RCS depends on the geometry of the object compared to the radar wavelength. The Rayleigh region, shown in Figure C-7, is where the wavelength is large compared to the object's dimensions. In the case of a sphere, this is the area where $2\pi a/\lambda \ll 1$ [31]. The

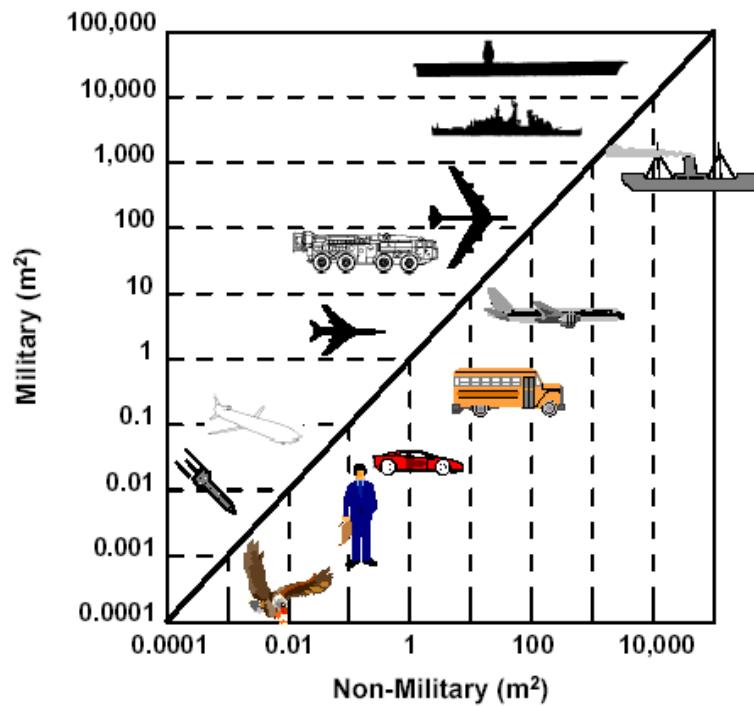


Figure C-6: Approximate RCS Values. Typical average RCS values for aircraft range anywhere from -70 dBsm to 25 dBsm (10^{-7} m^2 to 300 m^2). This figure shows comparative RCS values for birds (around 0.01 m^2) all the way up to aircraft carriers ($100,000 \text{ m}^2$).

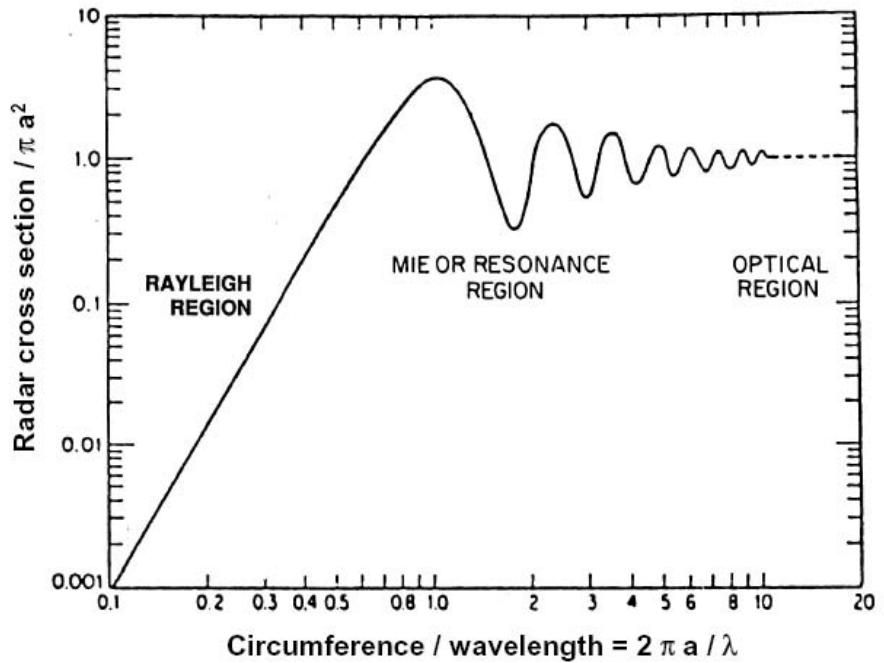


Figure C-7: RCS of a Sphere. This figure shows the normalized RCS of a sphere as the circumference (measured in wavelengths) is increased. The Rayleigh region on the left of the figure is where the wavelength is large compared to the object's dimensions. The optical region on the right is where the wavelength is very small compared to the object's dimensions, and the area in between is the resonance region, where the dimensions and wavelength are similar.

region was named after Lord Rayleigh, who observed similar scattering in 1871 while studying the scattering of light by microscopic particles [31]. Within the Rayleigh region, $RCS \propto f^4$, where f is the radar's frequency, and the volume of the target has a much larger effect than does the shape. Rain normally falls under this category.

The opposite side of Figure C-7 shows the optical region, where the RCS is heavily dependent on the shape of the target rather than the volume [31]. Changes in frequency or aspect angle result in large changes in RCS. In the optical region, the wavelength is very small compared to the dimensions of the object, and for a sphere, $2\pi a/\lambda \gg 1$. Scattering from aircraft or ships normally fall in this category. The area in between the Rayleigh and optical regions is called the resonance region, where the object's dimensions and the radar wavelength are similar. In this case, geometry and volume play roughly equal roles in determining RCS, which fluctuates as a function of frequency in the case of the sphere. The reason for this fluctuation is that two separate waves interfere in constructive or destructive ways. The first wave is the echo from the front face of the sphere. The second wave is called the creeping wave, which travels behind the sphere and heads for the radar to interfere with the first wave [31]. Figure C-8 [30] shows the backscatter produced by a short-duration pulse radar with both waves clearly depicted. In this case, the sphere's radius is equal to the radar wavelength, which indicates the resonance region. The creeping wave appears a short time after the first wave due to the distance traveled around the sphere.

C.2.2 Complex Targets

Viewing aspect and frequency are significant factors when determining the RCS for a complex target such as missiles or aircraft. Each object is made up of numerous parts that produce individual echoes. In the end, the radar sees the combination of all of these scatterers. Further complexity results from interactions between individual scatterers and differences in scatter properties [31].

For a complex target such as a guided missile, there exist three distinct contributions to the RCS: structure, propulsion, and avionics. These contributions are illustrated in Figure C-9 [30]. Structural contributions include control surfaces, access panels, the shape of the airframe, and even small fasteners. Propulsion contributions include the inlet, exhaust, and any heat

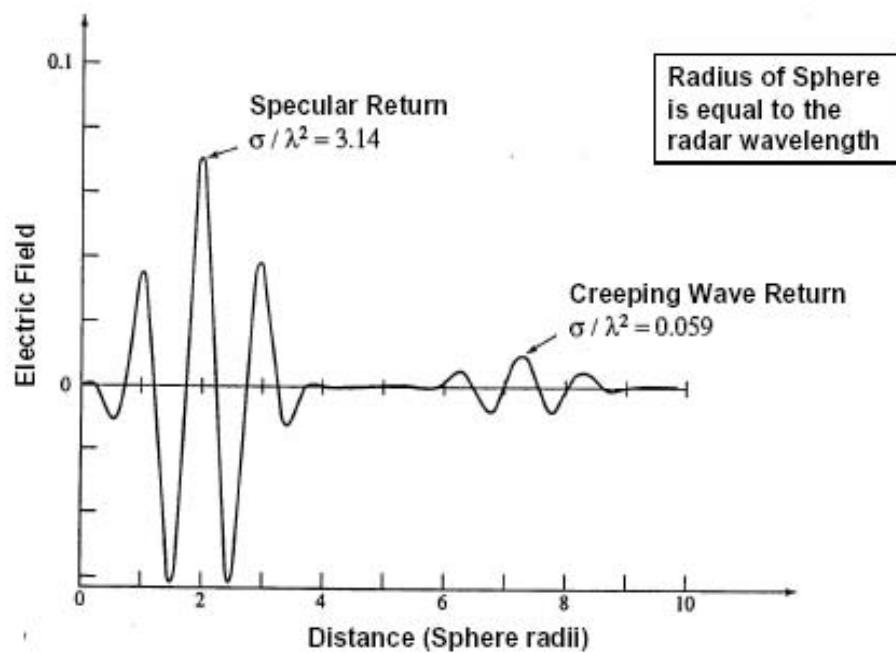


Figure C-8: Sphere Backscatter from Short Pulse Radar. Backscatter is produced by a short-duration pulse radar. The reason for frequency fluctuation is that two separate waves interfere in constructive or destructive ways. The first wave is the echo from the front face of the sphere. The second wave is called the creeping wave, which travels behind the sphere and heads for the radar to interfere with the first wave.

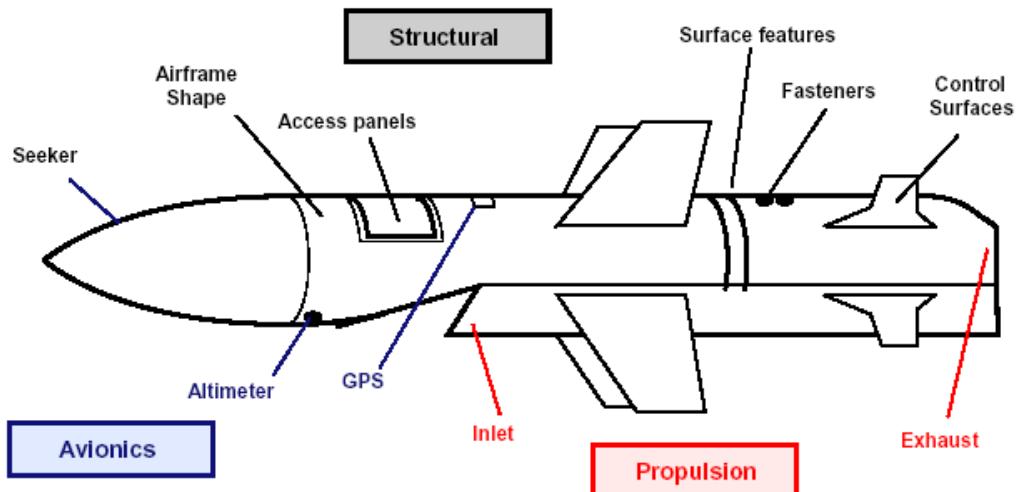


Figure C-9: Components of Target RCS. Complex targets have three distinct contributions to the RCS. Structural contributions include control surfaces, access panels, the shape of the airframe, and even small fasteners. Propulsion contributions include the inlet, exhaust, and any heat emissions. Avionics include items such as a GPS or altimeter.

emissions. Finally, avionics include items such as a GPS or altimeter. An examination of the colossal number of contributors to RCS on a typical aircraft should provoke a great deal of admiration for those who are able to minimize these scatterers and achieve stealth capability.

Older aircraft that have been studied in the past provide an opportunity to analyze RCS patterns in more detail. The Army Air Force's T-33 Shooting Star (Figure C-10 [27]) was the first jet fighter trainer, introduced in 1944 and exported to more than 20 countries. Figure C-11 [27] shows its RCS as a function of aspect angle, where 0 degrees is looking head on. It is clear that the largest spikes occur at 90 degrees and 270 degrees (for the two wings) with smaller spikes at 0 and 180 degrees for the nose and tail. Some find this view of RCS a little confusing and not very explanatory. Thus, the polar view of a target's RCS has become the main tool with which to analyze the scatter properties of a target. The commonly used example of a B-26 twin-engine bomber is shown in Figure C-12 [31]. This view succinctly displays the RCS pattern of the B-26 with the aircraft's frame superimposed in the center to provide an excellent comparison of RCS values and target geometry.



Figure C-10: T-33 Shooting Star. The T-33 is an older aircraft that has been studied in the past and thus provides a good opportunity to analyze RCS patterns in more detail. This specific plane was the first jet fighter trainer, introduced in 1944 and exported to more than 20 countries.

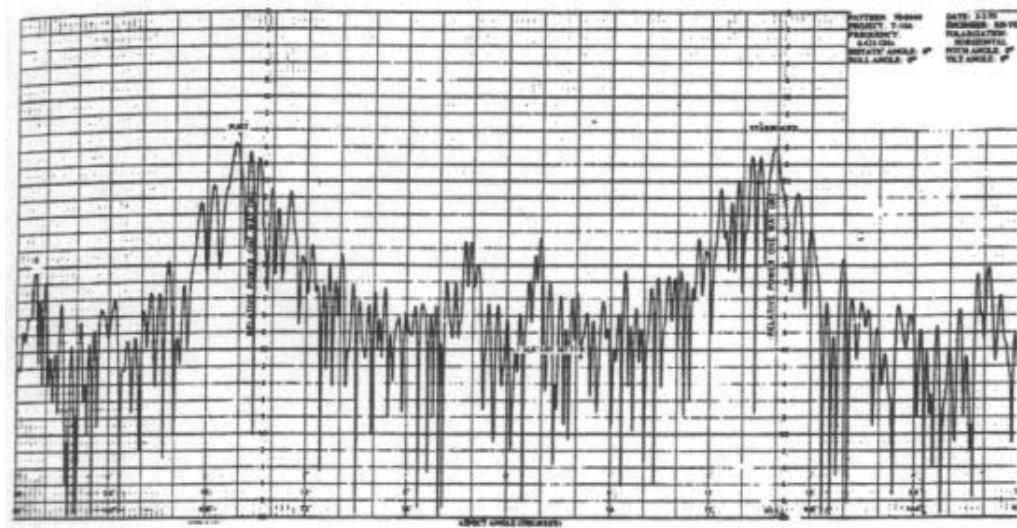


Figure C-11: T-33 RCS Pattern. The plot maps RCS versus aspect angle. The pattern clearly shows large spikes at 90 degrees and 270 degrees for the two wings and smaller peaks at 0 degrees and 180 degrees for the nose and tail.

C.2.3 Calculating RCS

RCS can be determined in two ways. It can be measured, or it can be predicted. Obviously, the first choice would be to measure it by using the actual target in flight tests where radar waves are transmitted and received by a ground station at a variety of aircraft depression and azimuth angles. Unfortunately, access to a particular aircraft is relatively rare. Thus, full-scale models of the aircraft are constructed and used to conduct RCS scatter tests. Styrofoam can be used in columns to support the aircraft in the test chamber since the dielectric properties of styrofoam are similar to those of free space [30]. The aircraft could also be mounted with metal pylons that are shaped to reduce radar echoes, and background subtraction techniques are utilized to produce a clear RCS pattern [30]. Scaled models can also be used, although scaling the measurements can become difficult or make for rough interpolations. For example, if the full-scale model would be measured at frequency f and the scale factor is S , then the smaller model should be measured at a frequency of $S \times f$ [30]. Other quantities, such as length, time, and conductivity have to be scaled by certain factors before accurate measurements can take place.

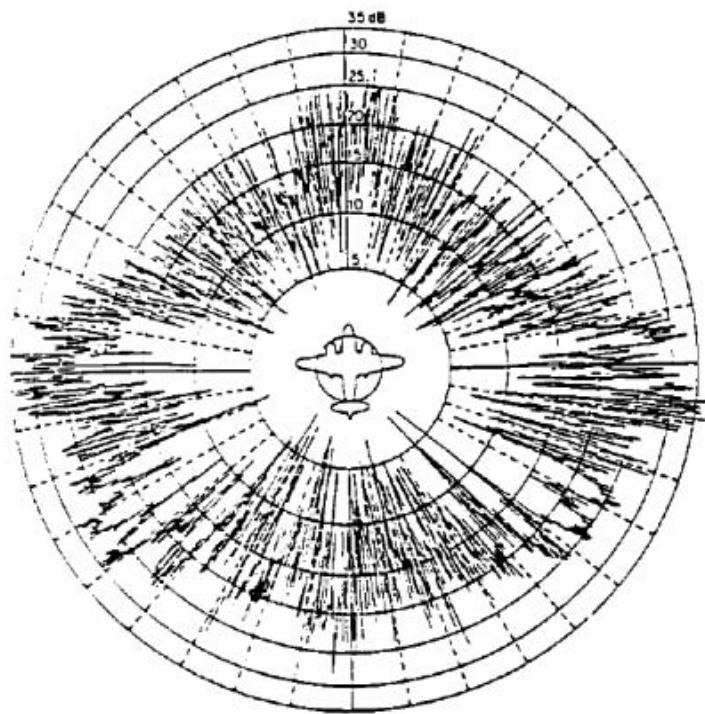


Figure C-12: B-26 RCS Polar Plot. Polar plots such as this one are more explanatory and increasingly are the main tool with which to analyze the scatter properties of a target. The most common RCS polar plot example is the B-26 twin-engine bomber.

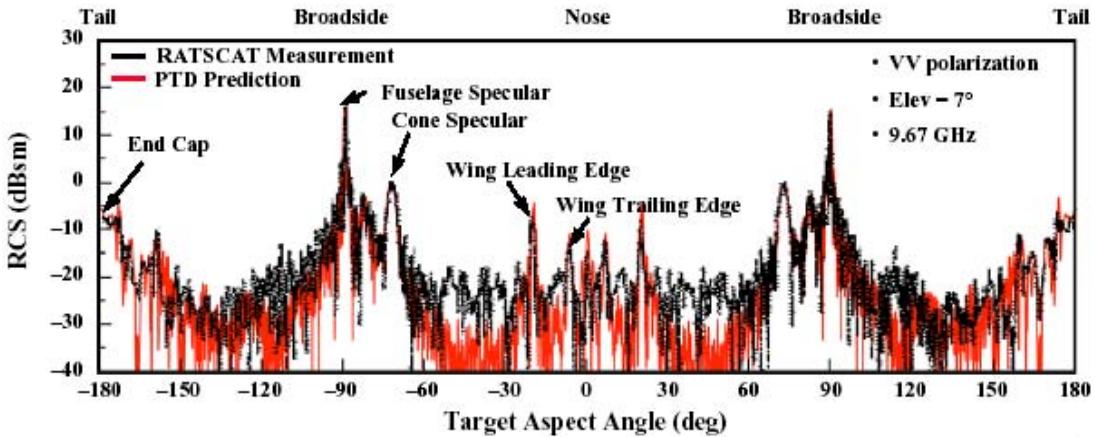


Figure C-13: Measured and Calculated RCS of JGAM. Clearly, individual RCS values for different aspect angles vary widely for different RCS prediction methods. Even averaged RCS curves superimposed on this figure would differ greatly at certain sets of angles.

RCS can also be predicted. Some texts on electromagnetic scattering give the following definition of RCS [31]:

$$\sigma = \frac{\text{power reflected toward source/unit solid angle}}{\text{incident power density}/4\pi} = 4\pi R^2 \frac{|E_r|^2}{|E_i|^2}, \quad (\text{C.7})$$

where R is range, E_r is the received electric field strength of the target echo, and E_i is the electric field strength incident on the target. In reality, this equation is not perfect, nor are the quantities contained in it easy to measure. One approach to predicting RCS involves high frequency approximations or exact numerical approaches [30]. High frequency approximations use the physical theory of diffraction to estimate the RCS, which reduces the computational demands. However, this method neglects multiple scattering and is applicable only to large, smooth geometries [30]. Numerical approaches use the method of moments to derive an exact formulation for the RCS of an object. The only problem is the intense computational requirements demanded by this method [30]. Figure C-13 [30] depicts both the measured and calculated RCS for the Johnson Generic Aircraft Model (JGAM). Clearly, individual RCS values for different aspect angles vary widely for each method. Even averaged RCS curves superimposed on Figure C-13 would differ greatly at certain sets of angles. RCS determination

is undoubtedly a field of great complexity, lacking perfect measurement and prediction tools. Many iterations with various models and methods must be performed before settling on a certain RCS.

C.2.4 Fluctuations

As previously stated, complex targets are composed of numerous parts which all scatter individually with different phases and amplitudes. At the radar receiver, all of these scatterers are summed to form a resultant phase and amplitude. Any small changes in the depression or azimuth angles of a target, as viewed by the radar, can result in major changes to the target RCS, described by that summed phase and amplitude. These changes are called fluctuations, and are exceedingly difficult to model accurately.

Peter Swerling described four statistical models to represent target fluctuations by utilizing a probability density function (pdf), which gives the probability of having a certain RCS in a certain range [31]. The first Swerling model assumes that the target echo pulses received by the radar have constant amplitude throughout an entire scan but are uncorrelated from scan to scan [31]. This is called a slow fluctuation, and the applicable pdf is shown below in Equation C.8 [31]:

$$p(\sigma) = \frac{1}{\sigma_{av}} \exp\left(-\frac{\sigma}{\sigma_{av}}\right) \quad \sigma \geq 0, \quad (\text{C.8})$$

where σ_{av} is the average over all values of target RCS. The second Swerling model is identical to the first, except that fluctuations are independent from pulse to pulse instead of scan to scan. This is called a fast fluctuation. The third Swerling model makes the same assumptions as the first one, but instead uses the following pdf:

$$p(\sigma) = \frac{4\sigma}{\sigma_{av}^2} \exp\left(-\frac{2\sigma}{\sigma_{av}}\right) \quad \sigma \geq 0. \quad (\text{C.9})$$

This pdf is supposedly better suited to targets with one large scatterer and numerous smaller scatterers [31]. The fourth and final Swerling model uses the same pdf in Equation C.9, but the fluctuation is pulse to pulse as in the second model.

Other target fluctuation models include the Chi-Square target model, the Rice pdf, and many others [31]. However, the most common models used to estimate RCS with fluctuations

are the Swerling models. In particular, the first Swerling model is the most widely used. Fluctuation modeling remains another complex area, and even one author admits that simply picking a low, conservative RCS that will be exceeded most of the time might be the best strategy since no existing analytical method is much more precise [31].

C.3 The Radar Equation

The radar equation relates the performance of a radar to its design parameters. Specifically, it connects the properties of the target, the characteristics of the radar, and the attributes of the propagation medium to obtain the range of the target and other things. It is mainly used to determine the maximum range at which a target can be detected and is often referred to as the radar range equation. However, the equation can also be very helpful in analyzing the relationships between different factors that affect radar performance. An understanding of these factors is paramount in designing a radar system to search for targets as well as knowing how to avoid detection.

If an antenna radiates uniformly in all directions (isotropically), the power density at a certain range R from the antenna can be calculated by the following equation:

$$\text{Power Density}_{\text{isotropic}} = \frac{P_t}{4\pi R^2}, \quad (\text{C.10})$$

where P_t is the peak transmitter power [30]. However, radars use directive antennas, which concentrate the power in a certain direction [31]. The term gain G_t is used to describe that increased power density in one direction as opposed to an isotropically radiated power density. Thus, the power density of a directive antenna is calculated as follows:

$$\text{Power Density}_{\text{directive}} = \frac{P_t G_t}{4\pi R^2}. \quad (\text{C.11})$$

The target itself is hit with these waveforms, which scatter in all directions. That fraction of the incident energy that is reradiated back in the direction of the radar is the echo signal, which is determined by the RCS σ of the target. The power density finally received by the radar is then multiplied by the effective area of the receiving antenna A_e to give the received power P_r

[30]:

$$P_r = \frac{P_t G_t}{4\pi R^2} \frac{\sigma A_e}{4\pi R^2} = \frac{P_t G_t \sigma A_e}{(4\pi)^2 R^4}. \quad (\text{C.12})$$

Two substitutions in Equation C.12 can then be made. From antenna theory, $G_t = 4\pi A_e / \lambda^2$, and thus $A_e = G_t \lambda^2 / 4\pi$ [31]. Also, to find the maximum detection range of the radar R_{\max} , the minimum detectable signal S_{\min} must be substituted for P_r , giving:

$$R_{\max} = \left[\frac{P_t G_t^2 \sigma \lambda^2}{(4\pi)^3 S_{\min}} \right]^{1/4}. \quad (\text{C.13})$$

Equation C.13 is the fundamental form of the radar range equation. However, many important factors must still be analyzed and included for the radar equation to accurately reflect radar performance.

One of the largest factors to influence radar signals is noise. As discussed in an earlier section, detection thresholds must be set at some value above the ever-present noise in the electromagnetic spectrum, and tradeoffs must occur between false alarms and missed detections as the threshold is moved. Many types of noise exist, to include solar or galactic noise, atmospheric noise, man-made noise, ground noise, and receiver noise. Figure C-14 [30] depicts all of these types of noise. The thermal noise power N generated by the agitation of electrons within the receiver is related to the noise bandwidth of the receiver B_n and the system noise temperature T_s [31] by the following equation:

$$N = k B_n T_s, \quad (\text{C.14})$$

where k is Boltzmann's constant, 1.38×10^{-23} joules / degree Kelvin. T_s can be broken down even further into the addition of three separate components. The antenna noise temperature T_a is the contribution from the antenna, which comes from the temperature of the sky and losses within the antenna. The RF noise temperature T_r is the contribution of RF components in the antenna and the receiver. Finally, the last contribution comes from the multiplication of the RF component loss L_r and the receiver temperature T_e . Thus, the system noise temperature is calculated as follows: $T_s = T_a + T_r + L_r T_e$ [30].

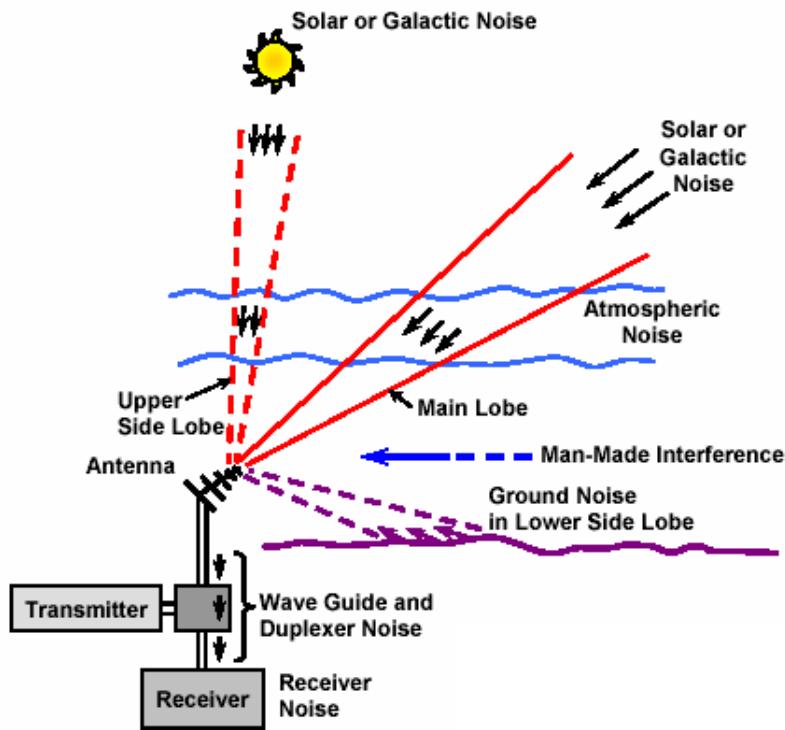


Figure C-14: Sources of Radar Noise. One of the largest factors to influence radar signals is noise. Detection thresholds must be set at some value above the ever-present noise in the electromagnetic spectrum, and tradeoffs must occur between false alarms and missed detections as the threshold is moved. Many types of noise exist, to include solar or galactic noise, atmospheric noise, man-made noise, ground noise, and receiver noise.

An important metric for measuring radar performance is the signal-to-noise ratio (S/N). It is essentially the ratio of the power received P_r to the power of the noise N , and it is the standard measure of a radar's ability to detect a target at a given range [30]. Thus $S/N = P_r/N$, and $P_r = N \cdot S/N$. Both transmit and receive losses are also significant when dealing with the radar equation. Loss examples include beam shape loss, where the target echo from a scanning radar is modulated by the shape of the antenna beam as it scans across the target. Another example is atmospheric attenuation loss, where the radar beam attenuates as it travels through the atmosphere both to the target and from it. The summation of all of these losses L can be inserted into the radar equation to provide more accuracy.

By inserting these new terms and substituting Equation C.14 into Equation C.13, the following track radar equation is found for S/N :

$$S/N = \frac{P_t G^2 \lambda^2 \sigma}{(4\pi)^3 R^4 k T_s B_n L}. \quad (\text{C.15})$$

Equation C.15 is called the track radar equation because the location of the target is assumed to be known, and the antenna is pointed directly towards the target. In track, the collective term $P_t A_e G$ is critical in designing a radar to a specific performance [30]. When the target's location is unknown, the radar must search a large volume of the sky to find it, and the resultant search radar equation is as follows:

$$S/N = \frac{P_{av} A_e t_s \sigma}{4\pi \Omega R^4 k T_s L}, \quad (\text{C.16})$$

where P_{av} is the average power, Ω is the solid angle searched, and t_s is the scan time for Ω , with all other terms having been previously defined. In search, $P_{av} A_e$ is the critical design parameter [30].

The equations presented here do not do justice to the complexity of radar detection. Many factors and terms have been excluded from the equations for simplicity. Nevertheless, they aid in understanding relationships between different parameters. From the radar design perspective, detailed analysis of the radar equation results in groups of parameters that should be increased to improve the performance of the radar. However, real world limitations on available power, aperture size, noise, and losses prevent the creation of super-performing radars,

and tradeoffs must be made, as always, between various parameters to achieve the desired performance. Conversely, the target has influence only over its RCS and its range from the radar (if radar location is certain). It cannot affect any other part of the radar equation. Thus, aircraft design, careful mission planning, and RCS management are the only tools available to the target seeking to avoid detection.

Appendix D

Sensors and Image Quality

The first job of a penetrating UAV is to travel deep inside enemy territory and avoid detection. Its second job might be to get a good image of the target it was sent to identify or locate. A variety of sensors exist for this purpose, with each sensor having specific advantages and disadvantages. To some extent, the image that is attained can also be evaluated by examining attributes such as range from the target, resolution, and other characteristics. This appendix will explore modern sensors and their capabilities.

D.1 Sensors

The simplest type of sensor is a camera. It uses mirrors and lenses to take videos, electro-optical (EO) images, and infrared (IR) images. EO images are very similar to images a photographer would produce because they both depend on light. During the day, most terrain features are clearly visible with the help of the sun. However, the lack of light at night makes EO sensors quite useless. Also, clouds and inclement weather render EO sensors obsolete because of the way they reflect light. Simplicity seems to be the main advantage of EO sensors. IR images function during night and day. They pick up infrared radiation (heat) released from areas on the Earth's surface. Places such as cities or warm water currents would appear as bright spots on an IR image, and clouds would appear in varying shades of gray.

The Global Hawk UAV's Integrated Sensor Suite (ISS) includes an EO/IR sensor produced by Raytheon. The narrow field of view (FOV) camera patches together numerous small frames



Figure D-1: Wescam 14 EO/IR Sensor. The Predator UAV uses a real-time video system mounted in the turrets of the aircraft. The Wescam 14 EO/IR sensor installed on the Predator RQ-1A.

to form a single image [26]. The U-2 carries a high resolution line scanning camera with a 7-band multispectral capability. The Hunter and Predator UAVs use real-time video systems mounted in the turrets of the aircraft. Figure D-1 [25] depicts the Wescam 14 EO/IR sensor installed on the Predator RQ-1A. Figure D-2 [53] illustrates the reach and capabilities of the Predator's video and IR sensors with both top-down and head-on views. The sensor points sideways at a range of up to 6.5 nautical miles at an altitude of 15,000 ft to achieve an image resolution of one foot. EO/IR sensors are good choices for penetrating aircraft on clear days.

Synthetic Aperture Radar (SAR) provides the capability to penetrate clouds and inclement weather as well as functioning at nighttime. It produces an image like any other radar. The time from transmission of a pulse to receiving the target echo helps determine the range, and the width of the pulse helps determine the range resolution. However, SAR also produces relatively fine azimuth resolution, which depends upon a physically large antenna (such as several hundred meters long) to focus energy into a sharp beam, especially because of the lower frequencies of SAR as compared to optical systems [24]. Since aircraft cannot practically carry such large antennas, they can instead collect the data while flying a certain distance and then process the data as if it came from a physically long antenna. The distance flown during data collection is then known as the synthetic aperture, resulting in a narrow synthetic beamwidth

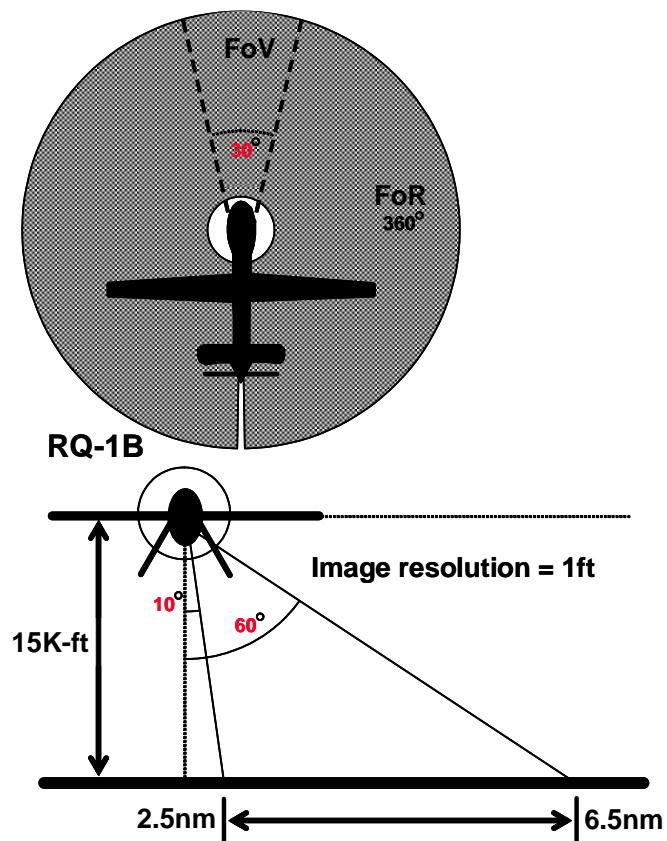


Figure D-2: Predator Video and IR Capability. This illustrates the reach and capabilities of the Predator's video and IR sensors with both top-down and head-on views. The sensor points sideways at a range of up to 6.5 nautical miles at an altitude of 15,000 ft to achieve an image resolution of one foot. EO/IR sensors are good choices for penetrating aircraft on clear days.

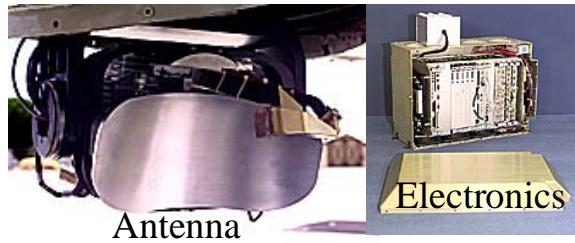


Figure D-3: Lynx SAR/GMTI. The Predator uses the Lynx Radar, which has a resolution of four inches in the spotlight mode and provides GMTI and change detection capabilities.

yielding finer resolution [24]. This relatively simplistic description serves only to provide a base understanding of SAR, which in actuality is a complicated topic whose design is exceedingly complex.

The U-2 carries a dedicated imaging SAR called Advanced SAR System (ASARS 2A), which is capable of one foot resolution [26]. Global Hawk's ISS is capable of various versions of SAR, such as spot, search, and Ground Moving Target Indicator (GMTI), also with resolutions down to one foot [26]. Tactical UAV Radar (TUAVR) is a 63 pound SAR/MTI radar for use on the Army's UAVs. It can provide one foot resolution in strip and spotlight modes and a moving target indicator (MTI) capability. The Predator uses the Lynx Radar, shown in Figure D-3 [25], which has a resolution of four inches in the spotlight mode and provides GMTI and change detection capabilities. Figure D-4 [53] depicts the reach of the Predator's SAR system. The most obvious difference between the Predator's EO/IR capability and its SAR capability is its decreased Field of Regard (FoR), meaning that every target being imaged by the Predator's SAR must be within a 40 degree swath on either side of the aircraft.

Other sensors in use but more associated with satellites are signals intelligence (SIGINT), which is the detection of any broadcast communication system. SIGINT is divided up into communications intelligence (COMINT) and electronic intelligence (ELINT) [39]. COMINT looks at the patterns, the source, and the content of message traffic. ELINT analyzes non-communication electronic transmissions, and is further divided into telemetry from missile tests (TELINT) and radar transmitters (RADINT). The U.S. operates four constellations of SIGINT satellites in geostationary and low-earth orbits, which have proven invaluable in past military

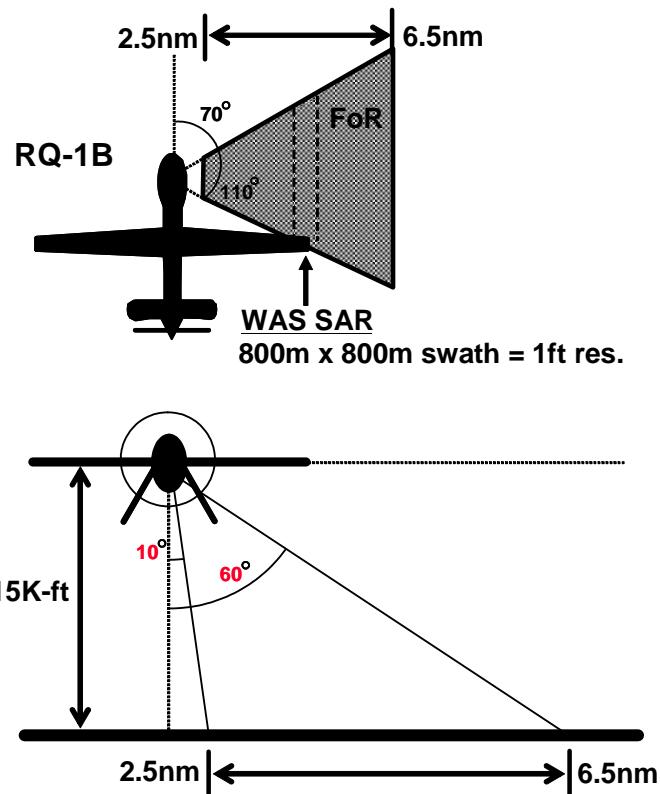


Figure D-4: Predator SAR Capability. This depicts the reach of the Predator's SAR system. The most obvious difference between the Predator's EO/IR capability and its SAR capability is its decreased Field of Regard (FoR), meaning that every target being imaged by the Predator's SAR must be within a 40 degree swath on either side of the aircraft.

experiences, such as providing the first warnings that the Iraqi invasion of Kuwait was likely in 1990 [39]. Other airborne assets also provide large contributions to SIGINT.

Other technologies include multispectral (tens of bands) and hyperspectral (hundreds of bands) imagery, which combine attributes of different sensors to form a target image that can divulge an increased amount of information [26]. U-2's sensor package utilizes a multispectral sensor, providing seven bands of imagery at high resolution [26]. Many technologies attempt to break through obstacles such as forest canopies or even enemy camouflage, such as UHF/VHF Foliage Penetration (FOPEN) and Light Detection and Ranging (LIDAR) [26]. Improvements in SAR are underway as well, with new systems using advanced algorithms to manipulate data and show precise changes in terrain with high resolution. Sensor hardware is also becoming more lightweight and efficient.

In general, sensor capabilities are excellent but still have a lot of room for improvement. With well-funded research into new technology, those improvements will come sooner rather than later. Unfortunately, sensors have become the "pacing item" for the cost of ISR UAVs [26]. Therefore, care must be taken to reduce cost growth and to plan UAV payloads more efficiently by combining sensor capabilities on one platform.

D.2 Image Quality

D.2.1 National Imagery Interpretability Rating Scale (NIIRS)

Taking an image is an excellent start, but interpreting the image is even more important. Image scale, resolution, and other image quality measures were never adequate enough to predict the interpretability of an image. More complex measures such as modulation transfer function (MTF) metrics could not clearly communicate the necessary information to the analysts [23]. Thus, a great deal of research was performed in the 1960s and 1970s that attempted to relate image quality to actual physical criteria. Up until the early 1970s, those efforts were to no avail, and one researcher concluded in 1970 that "no single measure of image quality is now available to satisfactorily predict (interpretation) performance" [23]. There are several reasons for this. Many tasks are needed to perform image interpretation, and differing quality did not uniformly affect those tasks. Also, some physical parameters interact differently with other parameters

than they do with interpretability. Finally, the studies that were performed analyzed only a small amount of targets and interpretation tasks, making generalization difficult [23].

In the early 1970s, a collaboration of government and contract workers developed the National Imagery Interpretability Rating Scale (NIIRS) for the Imagery Resolution And Reporting Standards (IRARS) Committee [23]. Image analysts first defined a standard set of interpretation tasks that they were commonly asked to perform. They were then given sets of images with pre-determined and varying quality and asked to describe the interpretation tasks that could be performed on the images. The original 10-level scale grew out of this study. Over the years, it has evolved many times, especially since many objects in the original NIIRS list became outdated [23]. NIIRS is now the standard scale used by image analysts, scientists, designers, and managers. It is task-based in that each NIIRS level implies different tasks that can be performed by an analyst [23]. With this tool, one can assess image quality as well as provide quantifiable means for expressing sensor system requirements. Four separate NIIRS scales have been developed: Radar NIIRS, Visible NIIRS, IR NIIRS, and Multispectral (MS) NIIRS [23]. These scales consist of 10 graduated levels (0 to 9) with every increase in level signifying a similar increase in the difficulty of the interpretation task required by the analyst.

Image analysts using a NIIRS 2 visible image, for example, should be able to detect large buildings. In contrast, image analysts should be able to identify automobiles as sedans or station wagons using a NIIRS 6 image. A full listing of every level in each type of NIIRS is included in Appendix E. Figure D-5 [39] shows an image of NIIRS 2. Large fields and road patterns can be seen. Figure D-6 [39] gives an example of a NIIRS 6 image. Clearly, the quality of the picture has improved. Types of automobiles can be distinguished and trails can be discerned. Finally, an example of NIIRS 8 is shown in Figure D-7 [39]. The quality of the NIIRS 8 image is evident. License plates can be identified along with windshield wipers on a vehicle. The development of this scale has enabled image analysts to use a common language. Additionally, it has significantly changed the manner in which requirements are set for different sensors and systems. For example, the Defense Airborne Reconnaissance Office has defined the sensor system performance goals for the Global Hawk and the Predator in terms of NIIRS [23]. System developers must now work to achieve the user's desired level of NIIRS performance.



Figure D-5: NIIRS Level 2 Image. This picture is an example of a NIIRS 2 image. Large fields and road patterns can be seen, which is typical of this type of image.



Figure D-6: NIIRS Level 6 Image. Clearly, the quality of this NIIRS 6 picture has improved over the NIIRS 2 example. Types of automobiles can be distinguished and trails can be discerned.

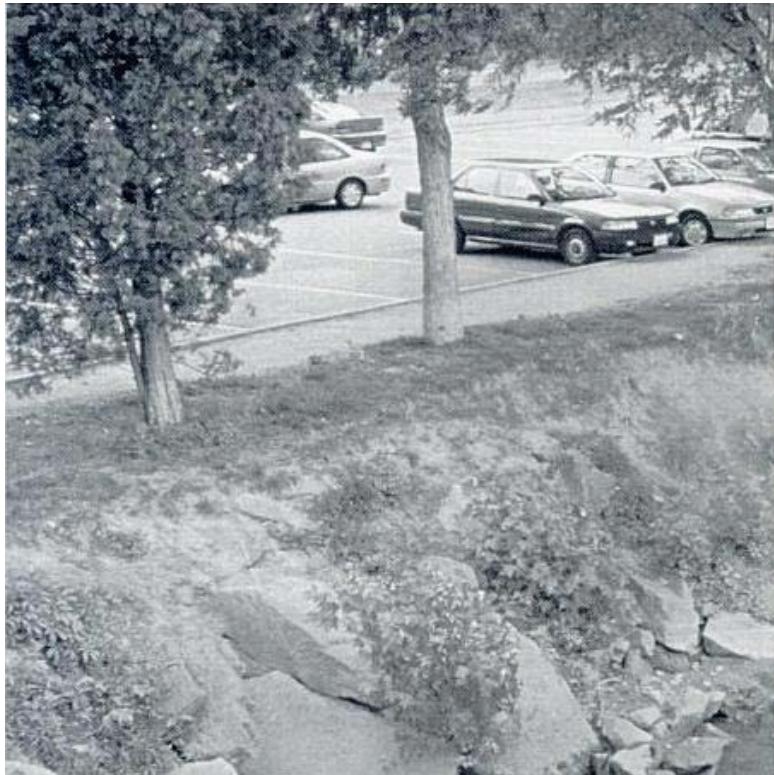


Figure D-7: NIIRS Level 8 Image. The quality of the NIIRS 8 image is quite evident. License plates can be identified along with windshield wipers on a vehicle

D.2.2 General Image Quality Equation (GIQE)

The widespread use of NIIRS by sensor system developers leads logically to the necessity of a method or technique that will accurately predict a NIIRS level based on sensor attributes before the actual sensor is built. Developed in the 1980s but not formally released until 1994, the General Image Quality Equation (GIQE) predicts NIIRS as a function of predicted image scale, sharpness or resolution, and SNR [22]. The GIQE was initially developed under the IRARS Committee and used a regression modeling approach. Ten image analysts conferred NIIRS ratings on samples of EO imagery. The characteristics of those samples were then used to develop a regression model that predicted NIIRS as a function of perceptual quality attributes of scale, resolution and sharpness, contrast, and noise [22]. The GIQE was released to the UAV community in 1994 but has undergone changes and improvements since that time.

The terms in the GIQE come from earlier research that related the quality of an image to its interpretability, and these terms account for all physical quality parameters that have been found to affect image interpretability. The original form of the GIQE for EO systems is as follows [22]:

$$NIIRS = 11.81 + 3.32 \log_{10} \left(\frac{RER_{GM}}{GSD_{GM}} \right) - (1.48H_{GM}) - \left(\frac{G}{SNR} \right). \quad (D.1)$$

RER_{GM} is the geometric mean of the Relative Edge Response (RER), which relates to perceived sharpness or acutance of the image. GSD_{GM} is the geometric mean of the Ground Sampled Distance (GSD), which measures both scale and resolution. EO system post-processing techniques involve modulation transfer function compensation (MTFC), which increases edge response and noise [22]. Thus, other terms were added to the GIQE to account for these increases. H_{GM} is the geometric mean of the height of the overshoot due to edge sharpening. G is the noise gain due to edge sharpening, and the SNR provides for some modeling of contrast. The GIQE does not account for bandwidth compression (BWC), softcopy image output prediction, or inclement weather [22].

RER is the slope of the edge system response and is found by measuring two points that are 0.5 pixels from the edge using a normalization over the range of 0 to 1. GSD is found with

the following equation [22]:

$$GSD = \frac{\left(\left(\frac{\text{pixel pitch}}{\text{focal length}} \right) \times \text{slant range} \right)}{\cos(\text{look angle})}. \quad (\text{D.2})$$

Both GSD and RER are computed along the X and Y axes, and then the geometric means are found. H is found by measuring the maximum value over the range 1.0 to 3.0 pixels from the edge at 0.25 pixel increments. The mean of this term is also found using the values along the X and Y axes.

A large sample of 359 NIIRS-rated images was then used to validate the GIQE. Because of unexpectedly lower RER/GSD and G/SNR coefficients, an updated equation modified some parameters and used two separate conditions for RER [22]:

$$\begin{aligned} NIIRS &= 10.251 - a \log_{10}(GSD_{GM}) + b \log_{10}(RER_{GM}) \\ &\quad - (0.656 \times H) - \left(0.344 \times \frac{G}{SNR} \right) \\ RER &\geq 0.9 : a = 3.32 \text{ and } b = 1.559 \\ RER &< 0.9 : a = 3.16 \text{ and } b = 2.817. \end{aligned} \quad (\text{D.3})$$

The GIQE performed quite well in studies. Figure D-8 [22] displays the relationship between predicted and observed NIIRS using the GIQE. Clearly, it is an accurate predictor of NIIRS levels.

The approach used to develop the EO GIQE was applied in a similar fashion to develop an IR GIQE, which is almost exactly the same as Equation D.3 except for an 0.5-NIIRS offset [23]. Research still continues in developing good image quality metrics and quantifiable tools for them, but the NIIRS/GIQE combination has proven to be extremely important in assessing and quantifying the interpretability of imagery collected during ISR missions. Although only EO and IR GIQEs have been developed, the principles involved can be applied to any sensor. In Chapter Two, this thesis presents a modification of Equation D.1 for use in simulations.

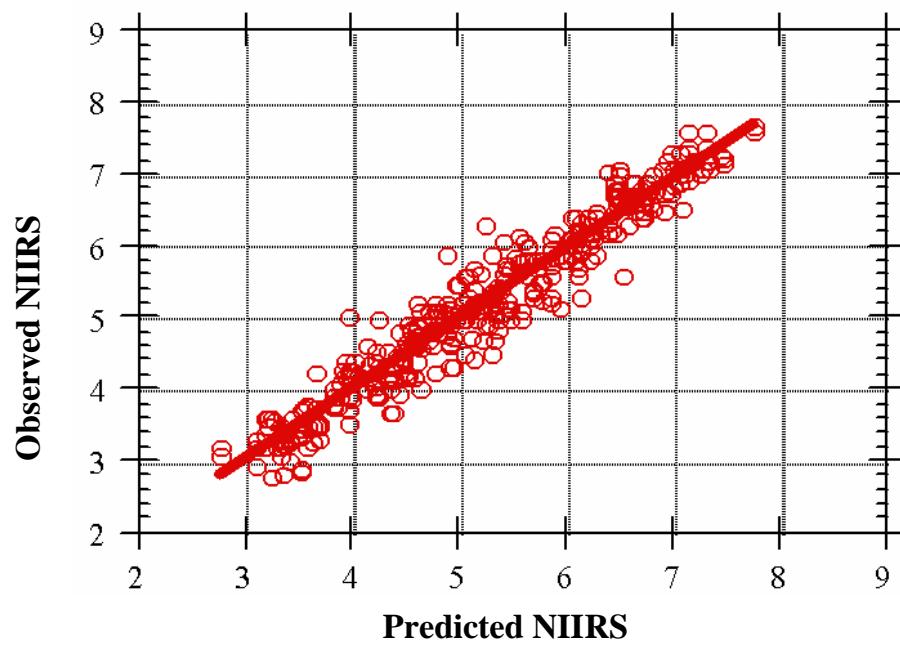


Figure D-8: Predicted and Observed NIIRS Values. The red circles represent the predicted NIIRS values according to the GIQE as compared to the observed NIIRS values by image analysts. The straight red line is the best fit line, showing that the predictions were clearly accurate.

Appendix E

National Image and Interpretability Rating Scales (NIIRS)

E.1 Visible NIIRS March 1994

Rating Level 1

- Detect a medium-sized port facility and/or distinguish between taxi-ways and runways at a large airfield.

Rating Level 2

- Detect large hangars at airfields.
- Detect large static radars (e.g., AN/FPS-85, COBRA DANE, PECHORA, HENHOUSE).
- Detect military training areas.
- Identify an SA-5 site based on road pattern and overall site configuration.
- Detect large buildings at a naval facility (e.g., warehouses, construction hall).
- Detect large buildings (e.g., hospitals, factories).

Rating Level 3

- Identify the wing configuration (e.g., straight, swept, delta) of all large aircraft (e.g., 707, CONCORD, BEAR, BLACKJACK).
- Identify radar and guidance areas at a SAM site by the configuration, mounds, and presence of concrete aprons.
- Detect a helipad by the configuration and markings.
- Detect the presence/absence of support vehicles at a mobile missile base.
- Identify a large surface ship in port by type (e.g., cruiser, auxiliary ship, noncombatant/merchant).
- Detect trains or strings of standard rolling stock on railroad tracks (not individual cars).

Rating Level 4

- Identify all large fighters by type (e.g., FENCER, FOXBAT, F-15, F-14).
- Detect the presence of large individual radar antennas (e.g., TALL KING).
- Identify, by general type, tracked vehicles, field artillery, large river crossing equipment, wheeled vehicles when in groups.
- Detect an open missile silo door.
- Determine the shape of the bow (pointed or blunt/rounded) on a medium-sized submarine (e.g., ROMEO, HAN, Type 209, CHARLIE II, ECHO II, VICTOR II/III).
- Identify individual tracks, rail pairs, control towers, switching points in rail yards.

Rating Level 5

- Distinguish between a MIDAS and a CANDID by the presence of refueling equipment (e.g., pedestal and wing pod).
- Identify radar as vehicle-mounted or trailer-mounted.
- Identify, by type, deployed tactical SSM systems (e.g., FROG, SS-21, SCUD).

- Distinguish between SS-25 mobile missile TEL and Missile Support Vans (MSVs) in a known support base, when not covered by camouflage.
- Rating Level 5 (cont.)
- Identify TOP STEER or TOP SAIL air surveillance radar on KIROV-, SOVREMENNY-, KIEV-, SLAVA-, MOSKVA-, KARA-, or KRESTA-II-class vessels.
- Identify individual rail cars by type (e.g., gondola, flat, box) and/or locomotives by type (e.g., steam, diesel).

Rating Level 6

- Distinguish between models of small/medium helicopters (e.g., HELIX A from HELIX B from HELIX C, HIND D from HIND E, HAZE A from HAZE B from HAZE C).
- Identify the shape of antennas on EW/GCI/ACQ radars as parabolic, parabolic with clipped corners or rectangular.
- Identify the spare tire on a medium-sized truck.
- Distinguish between SA-6, SA-11, and SA-17 missile airframes.
- Identify individual launcher covers (8) of vertically launched SA-N-6 on SLAVA-class vessels.
- Identify automobiles as sedans or station wagons.

Rating Level 7

- Identify fitments and fairings on a fighter-sized aircraft (e.g., FULCRUM, FOXHOUND).
- Identify ports, ladders, vents on electronics vans.
- Detect the mount for antitank guided missiles (e.g., SAGGER on BMP-1).
- Detect details of the silo door hinging mechanism on Type III-F, III-G, and II-H launch silos and Type III-X launch control silos.

- Identify the individual tubes of the RBU on KIROV-, KARA-, KRIVAK-class vessels.
- Identify individual rail ties.

Rating Level 8

- Identify the rivet lines on bomber aircraft.
- Detect horn-shaped and W-shaped antennas mounted atop BACKTRAP and BACKNET radars.
- Identify a hand-held SAM (e.g., SA-7/14, REDEYE, STINGER).
- Identify joints and welds on a TEL or TELAR.
- Detect winch cables on deck-mounted cranes.
- Identify windshield wipers on a vehicle.

Rating Level 9

- Differentiate cross-slot from single slot heads on aircraft skin panel fasteners.
- Identify small light-toned ceramic insulators that connect wires of an antenna canopy.
- Identify vehicle registration numbers (VRN) on trucks.
- Identify screws and bolts on missile components.
- Identify braid of ropes (1 to 3 inches in diameter).
- Detect individual spikes in railroad ties.

E.2 Radar NIIRS April 1999

Rating Level 1

- -Determine azimuth of main runway at a large airfield.
- -Detect lines of transportation, either road or rail, but do not distinguish between.

- -Detect a large vessel in open water.

Rating Level 2

- -Detect parallel taxiways.
- -Distinguish between forested areas and agricultural fields.
- -Detect very large defensive berm (e.g., Iraqi defense during DESERT SHIELD/STORM).
- -Detect known ICBM facility.
- -Detect large freighters or tankers at a known civilian port facility.

Rating Level 3

- -Identify the basic functional areas of an airfield or air base (e.g., hangars, weapons storage, POL storage, passenger terminals).
- -Detect multiple wings of large buildings.
- -Detect known KRUG site.
- -Identify a barracks area based on pattern of buildings.
- -Detect Rail Transfer Point (RTP) at missile facility.
- -Distinguish between ships and floating dry docks.

Rating Level 4

- -Detect large fighter aircraft (e.g., FENCER, F-15, TORNADO) on a known parking apron.
- -Identify the wing configuration (e.g., straight, swept, delta) of large aircraft (e.g., BEAR, B1, TU-144 CHARGER).
- -Detect smokestacks in industrial facilities.
- -Detect large vertical lattice mast antenna.

- -Detect a convoy or preparations for deployment at a motor pool.
- -Detect coastal defense artillery battery based on location and dispersal pattern.
- -Identify square bow shape of ROPUCHA LST.
- -Detect SS/SSN at known port facility (e.g., KILO, TYPE 209, VICTOR III).

Rating Level 5

- -Distinguish between large bomber and cargo aircraft (e.g., BEAR v. CANDID).
- -Detect small helicopters (e.g., HOPLITE, IROQUOIS, ALOUETTE).
- -Distinguish between a large vertical mast antenna and a large power transmission tower.
- -Detect a battery of towed artillery (not revetted) based on deployment pattern.
- -Detect a LOW BLOW radar at a fixed SA-3 site based on position.
- -Distinguish between SS/SSN and SSBN.
- -Distinguish between DON AS and SMOLNYY AX based on superstructure configuration.

Rating Level 6

- -Identify large helicopters by type (e.g., HALO, HOOK, SUPER FRELON, SEA STALION).
- -Identify wing configuration of small fighter aircraft as swept, delta or straight (e.g., ALPHA JET, MIRAGE 2000, FROGFOOT).
- -Detect cargo on a railroad flatcar.
- -Detect deployed mast-mounted TWIN EAR B antenna.
- -Identify a single vehicle as a large truck (e.g., KRAZ-255, M939).
- -Distinguish between wheeled and tracked vehicles in garrison based on vehicle size and dimension.

- -Detect a warhead van at a known Rail Transfer Point (RTP).
- -Identify SS by class (e.g., KILO, DAPHNE, TYPE 209).
- -Identify bow gun on a destroyer.

Rating Level 7

- -Identify medium helicopters by type (e.g., HIP, HIND, PUMA).
- -Identify fighter aircraft by type (e.g., FENCER, TORNADO, KFIR, MIRAGE).
- -Detect the break between cab and trailer on tractor-trailer truck.
- -Distinguish between electronic van trailers (without tractor) and van trucks in garrison.
- -Distinguish between THIN SKIN A (trailer mounted) and THIN SKIN B (truck mounted).
- -Determine general function of engineering equipment when in garrison (e.g., bridge sections, boats, earth-movers and mine-laying/clearing).
- -Distinguish between a turreted, tracked APC and a medium tank by size and configuration (e.g., BMP 1/2 vs. T-72).
- -Distinguish between C-802 SACCADÉ missile launcher and radar vehicle when in garrison.
- -Determine if an SA-3 missile launcher is loaded or empty.
- -Distinguish between OSA-I and OSA-II PCFGs based on missile canister shape.
- -Identify closed missile hatches on a DELTA IV SSBN.

Rating Level 8

- -Identify small helicopters by type (e.g., HOPLITE, IROQUOIS, ALOUETTE).
- -Detect individual rail ties.
- -Detect the vertical ribs on the sail of a LONG TRACK radar.

- -Distinguish between PMP center sections and ramp sections when mounted on KRAZ-255 truck.
- -Distinguish between 2S6 and ZSU-23/4 SPAA guns by overall configuration.
- -Distinguish between CSSC-2 SILKWORM and CSSC-3 SEERSUCKER missiles.
- -Determine the location of the running light on the stern fin of DELTA III.
- -Identify the individual RBU tubes on surface combatants (e.g., KIROV CGN, KARA CG, KRIVAK FFG).

Rating Level 9

- -Distinguish between antenna configurations on aircraft models (e.g., CURL A vs. CURL B, HIP vs. HIP C).
- -Distinguish between models of fighter aircraft (e.g., FLANKER B-C, F-15 A-E).
- -Identify feedhorn on THIN SKIN B when not operational.
- -Detect both gun tubes on 2S6 SPAA gun.
- -Detect exhaust nozzle on solid fuel booster packs on SA-5 missile.
- -Detect gun barrels on PHALANX CIWS.

E.3 Infrared NIIRS April 1996

Rating Level 1

- Distinguish between runways and taxiways on the basis of size, configuration or pattern at a large airfield.
- Detect a large (e.g., greater than 1 square kilometer) cleared area in dense forest.
- Detect large ocean-going vessels (e.g., aircraft carrier, super-tanker, KIROV) in open water.

- Detect large areas (e.g., greater than 1 square kilometer) of marsh/swamp.

Rating Level 2

- Detect large aircraft (e.g., C-141, 707, BEAR, CANDID, CLASSIC).
- Detect individual large buildings (e.g., hospitals, factories) in an urban area.
- Distinguish between densely wooded, sparsely wooded and open fields.
- Identify an SS-25 base by the pattern of buildings and roads.
- Distinguish between naval and commercial port facilities based on type and configuration of large functional areas.

Rating Level 3

- Distinguish between large (e.g., C-141, 707, BEAR, A300 AIRBUS) and small aircraft (e.g., A-4, FISHBED, L-39).
- Identify individual thermally active flues running between the boiler hall and smoke stacks at a thermal power plant.
- Detect a large air warning radar site based on the presence of mounds, revetments and security fencing.
- Detect a driver training track at a ground forces garrison.
- Identify individual functional areas (e.g., launch sites, electronics area, support area, missile handling area) of an SA-5 launch complex.
- Distinguish between large (e.g., greater than 200 meter) freighters and tankers.

Rating Level 4

- Identify the wing configuration of small fighter aircraft (e.g., FROGFOOT, F- 16, FISHBED).
- Detect a small (e.g., 50 meter square) electrical transformer yard in an urban area.

- Detect large (e.g., greater than 10 meter diameter) environmental domes at an electronics facility.
- Detect individual thermally active vehicles in garrison.
- Detect thermally active SS-25 MSV's in garrison.
- Identify individual closed cargo hold hatches on large merchant ships.

Rating Level 5

- Distinguish between single-tail (e.g., FLOGGER, F-16, TORNADO) and twin-tailed (e.g., F-15, FLANKER, FOXBAT) fighters.
- Identify outdoor tennis courts.
- Identify the metal lattice structure of large (e.g. approximately 75 meter) radio relay towers.
- Detect armored vehicles in a revetment.
- Detect a deployed TET (transportable electronics tower) at an SA-10 site.
- Identify the stack shape (e.g., square, round, oval) on large (e.g., greater than 200 meter) merchant ships.

Rating Level 6

- Detect wing-mounted stores (i.e., ASM, bombs) protruding from the wings of large bombers (e.g., B-52, BEAR, Badger).
- Identify individual thermally active engine vents atop diesel locomotives.
- Distinguish between a FIX FOUR and FIX SIX site based on antenna pattern and spacing.
- Distinguish between thermally active tanks and APCs.
- Distinguish between a 2-rail and 4-rail SA-3 launcher.
- Identify missile tube hatches on submarines.

Rating Level 7

- Distinguish between ground attack and interceptor versions of the MIG-23 FLOGGER based on the shape of the nose.
- Identify automobiles as sedans or station wagons.
- Identify antenna dishes (less than 3 meters in diameter) on a radio relay tower.
- Identify the missile transfer crane on a SA-6 transloader.
- Distinguish between an SA-2/CSA-1 and a SCUD-B missile transporter when missiles are not loaded.
- Detect mooring cleats or bollards on piers.

Rating Level 8

- Identify the RAM aircoop on the dorsal spine of FISHBED J/K/L.
- Identify limbs (e.g., arms, legs) on an individual.
- Identify individual horizontal and vertical ribs on a radar antenna.
- Detect closed hatches on a tank turret.
- Distinguish between fuel and oxidizer Multi-System Propellant Transporters based on twin or single fitments on the front of the semi-trailer.
- Identify individual posts and rails on deck edge life rails.

Rating Level 9

- Identify access panels on fighter aircraft.
- Identify cargo (e.g., shovels, rakes, ladders) in an open-bed, light-duty truck.
- Distinguish between BIRDS EYE and BELL LACE antennas based on the presence or absence of small dipole elements.

- Identify turret hatch hinges on armored vehicles.
- Identify individual command guidance strip antennas on an SA-2/CSA-1 missile.
- Identify individual rungs on bulkhead mounted ladders.

E.4 Multispectral NIIRS January 2001

Rating Level 1

- -Detect field delineation based on vegetation differences.
- -Identify golf courses.
- -Detect natural surface runways (e.g., grass, bare earth) in open terrain.
- -Detect shoreline quays outside of a port facility in absence of ships.
- -Detect vessel (~300' length) at suspected narcotics transshipment point in open ocean.

Rating Level 2

- -Identify natural drainage pattern in tidal flats.
- -Detect extent of recent flooding based on soil moisture differences.
- -Detect windbreaks/hedgerows between fields.
- -Identify composition of runway/taxiways/parking aprons
(i.e., concrete, asphalt).
- -Detect wetland areas unsuitable for mobile Intercontinental Ballistic Missile (ICBM) traffic.
- -Detect submerged spoil field from dredging operations near a port facility.
- -Detect the presence of offshore surface vegetation along a beach-landing zone.

Rating Level 3

- -Distinguish between fallow and abandoned fields.
- -Detect effluent discharge into water from industrial facility.
- -Differentiate between open pit coal mine and a limestone quarry.
- -Identify large cargo aircraft as military or commercial based on paint color and/or scheme.
- -Detect rows of vehicles in a parking area.
- -Identify beach terrain suitable for amphibious landing operations.

Rating Level 4

- -Detect vegetation stress/aging in narcotics crops in reported eradication area.
- -Identify tennis court as being composed of grass, clay or rubber/composite.
- -Detect CC&D efforts (cut vegetation camouflage netting) at suspected coca processing facilities.
- -Identify azimuth markings (numbers) on runway.
- -Detect tanks and SP-guns in revetted positions.
- -Detect blast marks from mobile or silo based ICBM launches on concrete.
- -Detect presence of Sea-Land containers on a ship's deck.
- -Distinguish between coal and sand loaded on dumb barge.

Rating Level 5

- -Detect individual trees with indications of vegetation stress.
- -Identify color of unfurled sails on sailboats (20 to 30 feet in length).
- -Distinguish between military and civilian helicopters paint schemes.
- -Identify large ground forces equipment by type (e.g., tanks, SP-guns, ARVs).

- -Detect foxholes by ring of spoil outlining hole.
- -Detect the presence of sailors in formation on the deck of a large combatant during parade/review.
- -Identify colors of stripes/deck markings on major surface combatants.

Rating Level 6

- -Detect foot trail through tall grass.
- -Identify colored bands on rail cars.
- -Identify support personnel performing maintenance while on an aircraft.
- -Identify color of stack markings on merchant ships.
- -Detect the presence of an alga line on the dark surface of a submarine hull.

Rating Level 7

- -Identify medium farm animals by type (e.g., sheep, hogs, goats).
- -Distinguish road-safety signs by color.
- -Detect wing flaps and other articulating surfaces on fighter aircraft.
- -Distinguish between open and closed hatches on the turret of an MTLB chassis.
- -Detect windsock on helo deck.
- -Identify canvas covering the muzzles of an ADMG-630 gun.

Rating Level 8

- -Detect different species of trees based on color variation.
- -Detect a soccer ball in play at a sports field.
- -Determine if a person (with helmet) is in a fighter aircraft with the canopy open.

- -Identify turret mounted smoke dischargers on an M-1 MBT.
- -Detect color pattern on life-rings on surface vessels.

Rating Level 9

- -Identify rock layers in sedimentary outcropping.
- -Count cleats on a civilian ski boat.
- -Detect red/green wing lights on fighter aircraft.
- -Distinguish between armed soldiers and armed civilians based on clothing.
- -Identify the draft marks numbers on submarine bow.

Appendix F

Game Theory

The problem explored in this thesis involves finding the best trajectory for a penetrating UAV or other aircraft that travels deep inside the detection and engagement rings of modern SAMs in enemy territory to perform surveillance, take imagery, and return safely to friendly areas. Many researchers have studied trajectory optimization in great detail [19]. Methods used in the past include Mixed-Integer Linear Programming, which allows optimization applications in areas where variables must have integer values [14]. Probabilistic Road Maps work by combining large, pre-optimized routes with a series of small path segments to reach a goal [11]. Rapidly-exploring Random Trees uses only the small path segments to complete an entire route from start to end [12]. Voronoi diagrams utilize multiple connected edges positioned appropriately to allow for the integration smooth, flyable trajectories [13]. Studies using these methods have provided valuable insight, yet lack realistic scenarios or solutions. This thesis focuses on the specifics of penetrating a SAM ring, which necessitates analysis of other factors such as RCS management and image quality. In addition, any UAV path must take into account the capabilities and reach of the SAM. In a way, the UAV and SAM are engaging in a simple game. Specifically, they are engaging in a two person non-cooperative mixed zero-sum game. These terms are explained later in this appendix. The strategies of the UAV comprise the multitude of possible paths it could take to get a valid image. The strategies of the SAM may include firing the missile at different times to draw the UAV in close. With the use of game theory, a two-sided game such as this one can be scrutinized and solved to provide the most likely outcome.

F.1 A Brief History of Game Theory

A scattering of economists and mathematicians anticipated many of the ideas of game theory throughout the 19th and early 20th centuries [18]. The economists Augustin Cournot and Francis Ysidro Edgeworth published notable papers on the competition between producers and trading between individuals, respectively [21]. In studying the game of chess in 1913, the mathematician E. Zermelo discovered the basis for backwards induction, otherwise known as Zermelo's Theorem [21]. Emile Borel presented the first modern formulation of a mixed strategy, among other ideas, in the 1920s, and the mathematician John von Neumann later provided the basis for game theory in a paper published in 1928 [3]. However, the groundbreaking text on game theory was a collaboration between von Neumann and Oskar Morgenstern called *The Theory of Games and Economic Behavior*, published in 1944 [4].

Development of game theory continued with John Nash's publication of four papers from 1950-1953 [5, 6] that made vital contributions to non-cooperative game theory and bargaining theory, including proving the existence of a strategic equilibrium (now termed the Nash equilibrium) for non-cooperative games [18]. The publication of A.W. Tucker's lecture on the Prisoners' Dilemma in 1950 made game theory even more widely known. Notable game theorists who continued to contribute throughout the latter half of the 20th century include Aumann [7], Shapley [8], Selten [9], and Harsanyi [10]. The year of 1994 brought a large amount of public interest in the subject when the Nobel prize in Economic Science was awarded to John Nash, John C. Harsanyi and Reinhard Selten for their contributions to game theory. The subject is one of increasing complexity, and it remains one of the most powerful tools available for studying human interaction.

F.2 Characteristics of Game Theory

F.2.1 Definition, Application, and Rationality

Game theory describes interactions in which the outcomes depend on the individual strategies of two or more rational players whose motives are opposed or at least mixed [18]. For example, a person bidding at an auction is playing a game with the other bidders. A supermarket is playing a game with its customers and with other stores when it decides to sell cereal at a

certain price, and lawyers play games with each other when they decide on what type of defense or prosecution will be pursued [21]. Games can also be more serious in nature. Napoleon and Wellington were playing a game at the Battle of Waterloo, and Khrushchev and President Kennedy were playing a game during the Cuban missile crisis [21]. In reality, many situations in life can be described in terms of game theory.

Game theory has been applied in detail and practice to a variety of disciplines. The most important application is economics, which is about the allocation of scarce resources. Resources are scarce because more people want them than are able to receive them, which is a perfect setup for a game [21]. In the realm of political science, topics such as strategic voting and the choice of a party's political platform also incorporate game theoretic principles. Biology, social philosophy, and other social and behavioral sciences have used game theory in the past. In fact, some authors believe that the entire branch of social sciences is simply a subdiscipline of game theory [21].

However, a majority of game theory relies on rational interactions, and the world is certainly not a model of rationality. A rational person is one who seeks to maximize his rewards in any given situation. The rewards may be monetary or subjective in nature. The concept of rationality is extremely helpful in that it is much easier to predict rational behavior as opposed to irrational behavior [18], although rational models could be made more subtle to account for irrationality in every day life.

F.2.2 The Prisoners' Dilemma

Game theory is a tough concept for human beings to grasp because of the circular reasoning involved. For example, if John and Jane were playing a game, then John's choice of strategy, if rational, will depend on what Jane will do. However, Jane's strategy, if she is also rational, will depend on what John will do, which in turn depends on her own as of yet undecided strategy. This type of convoluted logic leads to some very interesting outcomes at times. Perhaps the best way to demonstrate game theory is with an actual example. The following case is called the Prisoners' Dilemma. In attempting to depict the difficulty of analyzing certain games, A.W. Tucker created this game while addressing a room full of psychologists at Stanford University in 1950 [18]. This particular game is probably one of the most scrutinized in all of game theory,

	Bob Confesses	Bob Remains Silent
Jack Confesses	10, 10	0, 20
Jack Remains Silent	20, 0	1, 1

Table F.1: Prisoners' Dilemma

and according to one author, it "could be the most influential one page in the social sciences in the latter half of the twentieth century" [18].

Suppose two burglars named Bob and Jack are captured near the scene of the crime and then placed in separate interrogation rooms by the police. Both Bob and Jack must decide whether to confess and implicate the other burglar or remain silent and confess to nothing. If neither of them confess, they will both serve one year jail sentences on charges of carrying a concealed weapon. If each of them confess and implicate the other burglar, they will both receive 10 year jail sentences. However, if only one of them confesses, he will be set free for collaborating with the authorities and the other burglar who remained silent will be given 20 years in jail.

The only two strategies available to the burglars are to confess or remain silent. The rewards (penalties, in this case) are the jail sentences. Being rational human beings, both Bob and Jack want to minimize that jail sentence. This game's attributes can be easily expressed in what is commonly called a payoff matrix, which shows the values of the payoffs (or penalties) according to each player's strategy.

Table F.1 shows Bob's strategies at the top with his associated payoffs being the number to the right in each box. Conversely, Jack's strategies are on the left, as are his payoffs in each box. Suppose Jack thinks like this: "If Bob confesses, then I get 10 years if I confess and 20 years if I remain silent, so my best strategy if Bob confesses is to confess. If Bob remains silent, then I get 0 years if I confess and 1 year if I remain silent, so confessing is my best option here as well. Thus, I will confess no matter what Bob does." However, Bob will reason this game out in exactly the same fashion. Thus, they will both confess and get 10 years each rather than acting irrationally, remaining silent, and getting only one year each. This result, in which individually rational actions result in worse outcomes for both players, has made an enormous impact in modern social science. The nuclear arms race, pollution, and road congestion are all real-world examples of this type of outcome. The simple Prisoners' Dilemma acts as a model

for these types of situations, which is why it remains such an important example [18].

F.2.3 Nomenclature

The end result of the Prisoners' Dilemma is a dominant strategy equilibrium. A dominant strategy is one that a player will choose regardless of the other players' strategies because it gives him maximum benefits regardless of the activities of others. If every player in a game utilizes a dominant strategy, then the outcome is called a dominant strategy equilibrium, which might also be a Nash equilibrium. However, a Nash equilibrium is not always a dominant strategy equilibrium. The Nash equilibrium applies only when a set of chosen strategies in a game exist such that no player can benefit by changing his strategy if the other players keep their strategies unchanged. What makes some games more interesting and problematic than others is the existence of multiple Nash equilibria [18].

Games also exist in many forms. The Prisoners' Dilemma is an example of a two person non-cooperative non-constant sum game. Games can be played by more than two players, and in fact, the majority of interactions in the world occur between more than two parties. The two burglars in the Prisoners' Dilemma had different goals, thus making that game non-cooperative. Many games exist in which the goals of each party are the same and thus cooperation is possible to achieve an outcome. Also, the payoffs in the Prisoners' Dilemma do not sum to a constant or to zero. In other words, Bob's losses (jail time) does not equal Jack's rewards (lack of jail time), nor do they sum to a certain constant. Thus, it is a non-constant sum game, which is more common in practical situations.

However, the losses and rewards are equal in a zero-sum (or constant-sum) game, which can be either pure or mixed. A pure zero-sum game is one in which each player chooses only one strategy. A mixed zero-sum game is one in which a player chooses among two or more strategies at random according to specific probabilities. The solution to a two person zero-sum game is quite clear and is called the maximin strategy. This occurs when each player chooses the strategy (or set of strategies) that maximizes their minimum payoff.

The problem of routing a penetrating UAV inside the detection radius of a SAM to take imagery and return to safety can be modeled as a two person non-cooperative mixed zero-sum game. The two person and non-cooperative parts are quite obvious. The problem is modeled

as a zero-sum game for simplicity because the UAV's goal (take an image, complete mission safely, etc..) are exactly the opposite of the SAM's goals (prevent image acquisition, shoot down UAV, etc...). It will also be shown in certain chapters that multiple Nash equilibria in the solution to this game will dictate mixed strategy solutions.

F.3 Calculating the Nash Equilibrium

Consider the payoff matrix $A = (a_{i,j})$ [20]. For a two person zero-sum game, one player chooses action i (out of n possible actions), and the other player chooses action j (out of m possible actions). Player one and player two receive payoffs of $a_{i,j}$ and $-a_{i,j}$, respectively. (\hat{i}, \hat{j}) is defined as the outcome of the game. A dominant strategy i^* for player one occurs when $a_{i,j} \leq a_{i^*,j}$ for all i and j . Similarly, a dominant strategy j^* for player two occurs when $a_{i,j} \geq a_{i,j^*}$ [20]. For a mixed zero-sum game, the players have mixed strategies, meaning that player one will select $\mathbf{x}^T = (x_1, x_2, \dots, x_n)$ where x_i corresponds to the probability that player one will choose action i . Player two will select $\mathbf{y}^T = (y_1, y_2, \dots, y_m)$ where y_j corresponds to the probability that player two will choose action j [20].

The goal of this game is for player one to maximize his average payoff and for player two to minimize player one's payoff (for a zero-sum game). In other words, player one wants to do the following [20]:

$$\text{maximize } \mathbf{x}^T A \mathbf{y} = \sum_i^n \sum_j^m x_i a_{i,j} y_j. \quad (\text{F.1})$$

Conversely, player two would like to attempt the following [20]:

$$\text{minimize } \mathbf{x}^T A \mathbf{y} = \sum_i^n \sum_j^m x_i a_{i,j} y_j. \quad (\text{F.2})$$

In a zero-sum game, only one payoff matrix is needed, and a mutual gain or loss in terms of payoff is not possible [20].

The outcome of a pure zero-sum game (\hat{i}, \hat{j}) is termed an equilibrium if $a_{i,\hat{j}} \leq a_{\hat{i},\hat{j}} \leq a_{\hat{i},j}$. For a mixed zero-sum game, the outcome $(\hat{\mathbf{x}}, \hat{\mathbf{y}})$ is a Nash equilibrium if the following is true

[20]:

$$\begin{aligned} \mathbf{x}^T A \hat{\mathbf{y}} &\leq \hat{\mathbf{x}}^T A \hat{\mathbf{y}} \leq \hat{\mathbf{x}}^T A \mathbf{y} \quad \text{for all } \mathbf{x} \in S_x \text{ and all } \mathbf{y} \in S_y \\ S_x &= \left\{ \mathbf{x} \mid 0 \leq x_i \text{ and } \sum_i x_i = 1 \right\} \\ S_y &= \left\{ \mathbf{y} \mid 0 \leq y_j \text{ and } \sum_j y_j = 1 \right\}. \end{aligned} \tag{F.3}$$

Some other properties exist for mixed zero-sum games [20]:

1. If $(\hat{\mathbf{x}}, \hat{\mathbf{y}})$ is a Nash equilibrium, then

$$\max_{\mathbf{x} \in S_x} \min_{\mathbf{y} \in S_y} \mathbf{x}^T A \mathbf{y} = \min_{\mathbf{y} \in S_y} \max_{\mathbf{x} \in S_x} \mathbf{x}^T A \mathbf{y} = \hat{\mathbf{x}}^T A \hat{\mathbf{y}}. \tag{F.4}$$

2. Stated differently, $(\hat{\mathbf{x}}, \hat{\mathbf{y}})$ is a Nash equilibrium if

$$\begin{aligned} \min_j \sum_i x_i a_{i,j} &\leq \min_j \sum_i \hat{x}_i a_{i,j} \quad \text{for all } \mathbf{x} \in S_x \\ \max_i \sum_j a_{i,j} y_i &\geq \max_i \sum_j a_{i,j} \hat{y}_j \quad \text{for all } \mathbf{y} \in S_y. \end{aligned} \tag{F.5}$$

3. There exists at least one mixed strategy Nash equilibrium for all A . When more than one equilibrium is found, each players' average payoff is independent of the equilibrium used.
4. If $A \geq 0$, then an equilibrium strategy for player one solves $\max_x v$ subject to $v \leq \sum_i x_i a_{i,j}$ for all j , with $\sum_i x_i = 1$ and $x_i \geq 0$. The optimal value of v corresponds to the average payoff to player one in equilibrium. If $x'_i = x_i/v$, then the following formula will equivalently find the Nash equilibrium:

$$\text{minimize } \sum_i x'_i \quad \text{subject to } \left\{ \begin{array}{l} \sum_i x'_i a_{i,j} \geq 1, \quad \text{for all } j \\ x'_i \geq 0, \quad \text{for all } i \end{array} \right\}. \tag{F.6}$$

Similarly, $\mathbf{y} = \mathbf{y}' / \sum_j y'_j$ is the equilibrium strategy for player two if $\mathbf{y}' = (y'_1, \dots, y'_n)$ is a solution to:

$$\text{maximize } \sum_j y'_j \quad \text{subject to } \left\{ \begin{array}{l} \sum_j a_{i,j} y'_j \leq 1, \quad \text{for all } i \\ y'_j \geq 0, \quad \text{for all } j \end{array} \right\}. \tag{F.7}$$

The payoff to player one is $v = 1/\sum_i x'_i = 1/\sum_j y'_j$.

A simple example might shed more light on this process. Suppose $A = \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}$. Equation F.6 can then be applied to yield a more specific problem definition in Equation F.8 [20]:

$$\text{minimize } x'_1 + x'_2 \quad \text{subject to} \quad \begin{cases} x'_1 + 2x'_2 \geq 1, & x'_1 \geq 0 \\ 2x'_1 + x'_2 \geq 1, & x'_2 \geq 0 \end{cases}. \quad (\text{F.8})$$

The solution to Equation F.8 is $x'_1 = x'_2 = \frac{1}{3}$, so the payoff to player one in equilibrium is $\frac{3}{2}$, and the Nash equilibrium is $(x_1, x_2) = \left(\frac{1}{2}, \frac{1}{2}\right)$.

It is never advantageous to take the first action in a zero-sum game [20]. However, if the first player uses mixed strategies, his disadvantages will be eliminated. Of course, the mixed zero-sum game between the UAV and the SAM require much larger payoff matrices and more complicated optimization routines than the illustrative example in Equation F.8. The effects of changing the values in the payoff matrix will also be harder to discern. However, the principles of finding the Nash equilibrium in using a game-theoretic solution will remain unchanged, and the increasing size of the payoff matrix and associated quantities will be overcome by the use of algorithms within computer simulations.

Appendix G

Program Code

The following code was written with Matlab computer programming software, Version 6.5, Release 13. It was run on a Dell desktop computer with a Pentium 4 processor, 1.8 GHz processor speed, and 1 GB of RAM using Microsoft Windows XP Professional, Version 2002.

G.1 CreateGraph.m

```
% CreateGraph  
%  
% 2Lt Daniel M Morales, USAF  
% MIT Lincoln Laboratory  
% 14 March 2005  
%  
% Description:  
% This program runs a Monte Carlo simulation of UAV paths while facing  
% uncertainty in the location of enemy SAM sites.  
%  
% Inputs:  
% None  
%  
% Outputs:
```

```

% None
%
% Functions called:
% UAVvsSAM
% ecr2gc
% FindPk
% ImageQuality
%
% Clear all variables and clear the Matlab workspace
clear all;
clc;
%
% Call UAVvsSAM, which optimizes the UAV's path given the SAM location
% using game theory techniques.
[PK,PQ,BestPathX,BestStratY,Paths,SAMecr,PixelLocations,UAVSpeed, ...
CompletionTimes,WaypointTimes,DetectionCutoff,DopplerCutoff, ...
MissileSpeed,OuterRadius,InnerRadius,StartRadius]=UAVvsSAM;
%
% Assign Paths, CompletionTimes, and WaypointTimes to just relate to
% the optimal path from UAVvsSAM, which lessens program run time.
Paths=Paths(BestPathX,:);
CompletionTimes=CompletionTimes(BestPathX,:);
WaypointTimes=WaypointTimes(BestPathX,:);
%
% Convert the locations of each waypoint from the geodetic coordinate
% system to the ECR system.
for i=1:size(PixelLocations,1)
    PixelLocations(i,:)=gc2ecr(PixelLocations(i,2),PixelLocations(i,1)...
    ,PixelLocations(i,3));
end
%
% Save the values from UAVvsSAM in the workspace
save 'uavvsam.mat'
%
% Used to load the previously saved values if UAVvsSAM not run
if(~exist('PK'))

```

```

load 'uavvsam.mat'
end

% Assign UAV Speed to 150 knots and convert to m/sec
UAVSpeed=0.07716667; % km/sec
UAVSpeed=UAVSpeed*1000; % m/sec

% NumLocations represents the number of randomly generated SAM locations
% for each average value of location uncertainty
NumLocations=100; % divisible by 4

TLE=1; % The first average uncertainty to be simulated, in miles
TLEend=100; % The last value of TLE, in miles
TLEstep=1; % The step value to proceed from first to last TLE, in miles

% Assign Probability of Kill and Picture Quality to a matrix that tabulates
% the TLE and the associated Pk and Pq values
PKTLE(1,1:2)=[0 PK];
PQTLE(1,1:2)=[0 PQ];
% Assign counter s to 2
s=2;

% Start 'while' loop at the initial TLE and end at TLEend
while TLE<=TLEend
    % Convert TLE to meters from miles
    TLE=sm2km(TLE)*1000;
    % Generate random x, y, and z values that average to be the current TLE
    DeltasXY=randn(NumLocations,1)*(TLE/4)+TLE;
    DeltasZ=randn(NumLocations,1)*10+30;
    % Distribute these random values amongst four different sets of SAM
    % locations for the x, y, and z coordinates (4 different quadrants)
    SAMlocations(1:(NumLocations/4),1)=SAMcr(1)+DeltasXY(1:...
        (NumLocations/4));
    SAMlocations(((NumLocations/4)+1):(NumLocations/2),1)=SAMcr(1)-...
        DeltasXY(((NumLocations/4)+1):(NumLocations/2));

```

```

SAMlocations(((NumLocations/2)+1):(NumLocations*3/4),1)=SAMecr(1)+...
DeltasXY(((NumLocations/2)+1):(NumLocations*3/4));
SAMlocations(((NumLocations*3/4)+1):NumLocations,1)=SAMecr(1)-...
DeltasXY(((NumLocations*3/4)+1):NumLocations);
SAMlocations(1:(NumLocations/4),2)=SAMecr(2)+DeltasXY(1:...
(NumLocations/4));
SAMlocations(((NumLocations/4)+1):(NumLocations/2),2)=SAMecr(2)-...
DeltasXY(((NumLocations/4)+1):(NumLocations/2));
SAMlocations(((NumLocations/2)+1):(NumLocations*3/4),2)=SAMecr(2)-...
DeltasXY(((NumLocations/2)+1):(NumLocations*3/4));
SAMlocations(((NumLocations*3/4)+1):NumLocations,2)=SAMecr(2)+...
DeltasXY(((NumLocations*3/4)+1):NumLocations);
SAMlocations(:,3)=SAMecr(3)+abs(DeltasZ);
% Start 'for' loop from 1 to NumLocations
for j=1:NumLocations
    % Load Azimuth / Depression angle look-up table for RCS
    load RCS
    % Calculate UAV location for all paths, updated every second, using
    % temporary variables and counters
    tempx=0;
    tempy=0;
    tempz=0;
    t=1;
    for x=2:size(Paths,2)
        % Find change in location from one waypoint to another waypoint
        Delta=PixelLocations(Paths(1,x),:)-PixelLocations(Paths(1,...
        x-1),:);
        % Use speed of UAV and Delta to calculate location of UAV at
        % each second
        if x==2

```

```

PixelLocations(Paths(1,x-1),1);
(Delta(1,1)*UAVSpeed*t)/norm(Delta);
Locations1(1,t)=PixelLocations(Paths(1,x-1),1)+((Delta...
(1,1)*UAVSpeed*t)/norm(Delta));
Locations2(1,t)=PixelLocations(Paths(1,x-1),2)+((Delta...
(1,2)*UAVSpeed*t)/norm(Delta));
Locations3(1,t)=PixelLocations(Paths(1,x-1),3)+((Delta(... 
1,3)*UAVSpeed*t)/norm(Delta));
t=t+1;
tempt=t-2;
else
if norm(Delta)==0
tempx=0;
tempy=0;
tempz=0;
else
tempx=Delta(1,1)*(UAVSpeed*(t-CompletionTimes(1,... 
x-1)))/norm(Delta);
tempy=Delta(1,2)*(UAVSpeed*(t-CompletionTimes(1,... 
x-1)))/norm(Delta);
tempz=Delta(1,3)*(UAVSpeed*(t-CompletionTimes(1,... 
x-1)))/norm(Delta);
end
Locations1(1,t)=PixelLocations(Paths(1,x-1),1)+tempx;
Locations2(1,t)=PixelLocations(Paths(1,x-1),2)+tempy;
Locations3(1,t)=PixelLocations(Paths(1,x-1),3)+tempz;
t=t+1;
tempt=t-1;
end
while t<CompletionTimes(1,x)

```

```

if norm(Delta)==0
Locations1(1,t)=Locations1(1,t-1);
Locations1(1,t)=Locations1(1,t-1);
Locations1(1,t)=Locations1(1,t-1);
else
Locations1(1,t)=PixelLocations(Paths(1,x-1),1)+tempx...
+((Delta(1,1)*UAVSpeed*(t-tempt))/norm(Delta));
Locations2(1,t)=PixelLocations(Paths(1,x-1),2)+tempy...
+((Delta(1,2)*UAVSpeed*(t-tempt))/norm(Delta));
Locations3(1,t)=PixelLocations(Paths(1,x-1),3)+tempz...
+((Delta(1,3)*UAVSpeed*(t-tempt))/norm(Delta));
end
t=t+1;
end
end

% Assign all location values after completion of mission to ending
% location
Locations1(1,(fix(CompletionTimes(1,end))+1):end)=Locations1(1,...
fix(CompletionTimes(1,end)));
Locations2(1,(fix(CompletionTimes(1,end))+1):end)=Locations2(1,...
fix(CompletionTimes(1,end)));
Locations3(1,(fix(CompletionTimes(1,end))+1):end)=Locations3(1,...
fix(CompletionTimes(1,end)));
% Convert SAM location from ECR system to geodetic system
[SAMlat,SAMlon,SAMalt]=ecr2gc(SAMlocations(j,:));
% Calculate Azimuth / Depression angle for each path, updated every
% second, using range and other information
for x=1:(size(Locations1,2)-1)
[Lat1,Lon1,Alt1]=ecr2gc([Locations1(1,x) Locations2(1,x) ...
Locations3(1,x)]);

```

```

[Lat2,Lon2,Alt2]=ecr2gc([Locations1(1,x+1) Locations2(1,...
x+1) Locations3(1,x+1)]);

Range(1,x)=sqrt(((Locations1(1,x)-SAMlocations(j,1))^2)+...
((Locations2(1,x)-SAMlocations(j,2))^2)+((Locations3...
(1,x)-SAMlocations(j,3))^2));

Dep(1,x)=fix(rad2deg(asin((Alt1-SAMalt)/Range(1,x))));

if SAMlon<Lon1
    Az(1,x)=fix(360+azimuth(Lat1,Lon1,Lat2,Lon2)-azimuth...
    (Lat1,Lon1,SAMlat,SAMlon));
else
    if azimuth(Lat1,Lon1,Lat2,Lon2)==0
        Az(1,x)=fix(360-azimuth(Lat1,Lon1,SAMlat,SAMlon));
    else
        Az(1,x)=fix(azimuth(Lat1,Lon1,Lat2,Lon2)-azimuth...
        (Lat1,Lon1,SAMlat,SAMlon));
    end
end
if Az(1,x)>360
    Az(1,x)=Az(1,x)-360;
end
Dep=abs(Dep);
Az=abs(Az);

% Extract RCS values for every UAV location from previously loaded
% RCS value matrix
for x=1:size(Dep,2)
    if Az(1,x)==360
        RCSpaths(1,x)=RCS(Dep(1,x)+1,Az(1,x)-1);
    else
        RCSpaths(1,x)=RCS(Dep(1,x)+1,Az(1,x)+1);
    end
end

```

```

end
end

% Calculate Signal to Noise Ratio based on 10 dB SNR for 0 dBsm
% target at 120 km range.

for x=1:size(RCSpaths,2)
SNR(1,x)=log10((2.0736*(10^30))*((10^(RCSpaths(1,x)-4.5))/...
(Range(1,x)^4)));
end

% Determine detections based on whether SNR is greater than or less
% than the specified detection cutoff, and disregard "detections"
% with Doppler velocity less than that specified cutoff.

for x=1:size(SNR,2)
if SNR(1,x)<DetectionCutoff
Detections(1,x)=0; % not detected
else
Detections(1,x)=1; % detected
end
if norm(UAVSpeed*cos(deg2rad(Az(1,x))))<DopplerCutoff % m/s
Detections(1,x)=0; % not detected
end
end

% Combine UAV paths with new SAM location to determine
% probabilities of kill

[ProbKill]=FindPk(Detections,SNR,Locations1,Locations2, ...
Locations3,UAVSpeed,MissileSpeed,SAMlocations(j,:), ...
OuterRadius,InnerRadius,Az);

% Assume SAM changes his strategy to maximize the Pk on the UAV
PKill=max(ProbKill);

% Find Pq using new SAM location
[PictureQuality]=ImageQuality(CompletionTimes,Range,Paths,RCS, ...

```

```

Az,Dep);
% Assign Pk and Pq to temporary variables
TempArray(j,1)=PKill;
TempArray(j,2)=PictureQuality;
end
% Set PK and PQ equal to the average of their values throughout all the
% locations generated
PK=sum(TempArray(:,1))/size(TempArray,1);
PQ=sum(TempArray(:,2))/size(TempArray,1);
% Convert TLE back to miles
TLE=km2sm(TLE/1000);
% Put Pk and Pq values in matrix form
PKTLE(s,1:2)=[TLE PK];
PQTLE(s,1:2)=[TLE PQ];
% Add 1 to counter s, and add the step value to TLE
s=s+1;
TLE=TLE+TLEstep;
end
% Save matrix values in case of computer failure / shut-down
save PKTLE
save PQTLE

```

G.2 UAVvsSAM.m

```

function [PK,PQ,BestPathX,BestStratY,Paths,SAMecr,PixelLocations,...
UAVSpeed,CompletionTimes,WaypointTimes,DetectionCutoff,DopplerCutoff...
,MissileSpeed,OuterRadius,InnerRadius,StartRadius]=UAVvsSAM
% UAVvsSAM
%
% 2Lt Daniel M Morales, USAF

```

```

% MIT Lincoln Laboratory
% 14 March 2005
%
% Description:
%
% This function uses game theory to optimize UAV paths with changing SAM
% location uncertainty.

%
% Inputs:
%
% None
%
% Outputs:
%
% PK from 0 to 1
%
% PQ from 0 to 9
%
% BestPathX unitless
%
% BestStratY unitless
%
% Paths unitless
%
% SAMecr m
%
% PixelLocations m, ECR coordinate system
%
% UAVSpeed km/s
%
% CompletionTimes s
%
% WaypointTimes s
%
% DetectionCutoff dB
%
% DopplerCutoff m/s
%
% MissileSpeed m/s
%
% OuterRadius km
%
% InnerRadius km
%
% StartRadius km
%
% Functions called:
%
% DtedData
%
% gc2ecr

```

```

% ecr2gc
% SetUpGrid
% CalculateDistances
% CreatePaths
% Infeasible
% Detect
% FindPk
% ImageQuality
% FindA
% Nash
% Roll
% Visualize
% Clear all variables from memory and workspace and close all figures
clear all;
clc;
close all;
% UAV characteristics
UAVSpeed=0.07716667; % km/sec (150 knots)
ImageRequirement=3; % deg
% RCSlow=-10; % dBsm
% RCShigh=0; % dBsm
% RCSwidth=30; % deg
% SAM characteristics
InnerRadius=20; % km
OuterRadius=50; % km
SAMlon=126.75; % deg East
SAMlat=38.75; % deg North
[SAMlat,temp]=DtedData(SAMlat,SAMlat,SAMlon,SAMlon); % m
SAMecr=gc2ecr(SAMlat,SAMlon,SAMalt); % m
MissileSpeed=0.99438; % km/sec (Mach 3)

```

```

DetectionCutoff=10; % dB
DopplerCutoff=1.388889; % m/sec (5 km/hour)
% Pixelized space characteristics
StepRadius=5; % km
StartRadius=OuterRadius+StepRadius; % km
AngleWidth=5; % deg, odd number
Height=20; % km
HeightSteps=10; % km
% Delineate waypoint locations
[PixelLocations]=SetUpGrid(InnerRadius,OuterRadius,StepRadius, ...
StartRadius,ImageRequirement,AngleWidth,SAMlon,SAMlat,SAMalt,Height, ...
HeightSteps);
Start=min(PixelLocations);
Stop=max(PixelLocations);
% Get elevation data for desired region
[Map,MapLegend]=DtedData(Start(2),Stop(2),Start(1),Stop(1));
% Convert latitude, longitude, and altitude into Earth Centered
% Rotating coordinates (ECR).
for i=1:size(PixelLocations,1)
PixelLocations(i,:)=gc2ecr(PixelLocations(i,2),PixelLocations(i,1), ...
PixelLocations(i,3));
end
% Calculate distances and times between waypoints
[Distances,WaypointTimes]=CalculateDistances(PixelLocations,UAVSpeed);
% Create set of possible paths and calculate path completion times
[Paths]=CreatePaths(AngleWidth,OuterRadius,InnerRadius,StepRadius, ...
HeightSteps,ImageRequirement);
% Search through set of possible paths and eliminate infeasible ones based
% on UAV turn limitations and terrain considerations.
[Paths,CompletionTimes]=Infeasible(Paths,PixelLocations,WaypointTimes, ...

```

```

Map,MapLegend,SAMlon,SAMlat,SAMalt);
% Use orientation and speed of UAV to calculate detections by SAM
[Dep,Az,Detections,SNR,Locations1,Locations2,Locations3,RCS,Range, ...
RCSPaths]=Detect(PixelLocations,Paths,UAVSpeed, ...
CompletionTimes,WaypointTimes,SAMecr,DetectionCutoff,DopplerCutoff);
% Convert PixelLocations back into geodetic system
for i=1:size(PixelLocations,1)
[PixelLocations(i,2),PixelLocations(i,1),PixelLocations(i,3)]=...
ecr2gc(PixelLocations(i,:));
end
% Combine UAV paths with SAM strategies to determine probabilities of kill
[ProbKill]=FindPk(Detections,SNR,Locations1,Locations2,Locations3, ...
UAVSpeed,MissileSpeed,SAMecr,OuterRadius,InnerRadius,Az);
% Calculate Image Quality for each path
[PictureQuality]=ImageQuality(CompletionTimes,Range,Paths,RCS,Az,Dep);
% Calculate the payoff matrix A
[A]=FindA(PictureQuality,CompletionTimes,ProbKill);
% Use game-theoretic principles to find Nash equilibrium
[X,UAVPayoff,PathProbsX,PathProbsY,SAMPayoff,Y]=Nash(A);
% Pick UAV strategy based on probable roll of dice
[BestPathX,BestStratY]=Roll(PathProbsX,PathProbsY);
% Assign Pk and Pq
PK=ProbKill(BestPathX,BestStratY);
PQ=PictureQuality(BestPathX,1);
% Draw picture of best path
Visualize(OuterRadius,InnerRadius,StepRadius,PixelLocations,AngleWidth, ...
SAMlon,SAMlat,BestPathX,Paths,HeightSteps);

```

G.3 DtedData.m

```
function [Map,MapLegend]=DtedData(LatStart,LatStop,LonStart,LonStop);
% DtedData
%
% 2Lt Daniel M Morales, USAF
% MIT Lincoln Laboratory
% 14 March 2005
%
% Description:
% This function produces the Digital Terrain Elevation Data (DTED) given
% specific latitude and longitude start and stop points
%
% Inputs:
% LatStart deg
% LatStop deg
% LonStart deg
% LonStop deg
%
% Outputs:
% Map unitless
% MapLegend unitless
%
% Functions called:
% dted
% Call dted function with specific directory to obtain elevation data
[Map,MapLegend] = dted(...  

'C:\Documents and Settings\moralesd\My Documents\Thesis Work\DTED Stuff',...
1,[LatStart LatStop],[LonStart LonStop]);
```

G.4 SetUpGrid.m

```
function [PixelLocations]=SetUpGrid(InnerRadius,OuterRadius, ...
StepRadius,StartRadius,ImageRequirement,AngleWidth,SAMlon,SAMlat, ...
SAMalt,Height,HeightSteps)
% SetUpGrid
%
% 2Lt Daniel M Morales, USAF
% MIT Lincoln Laboratory
% 14 March 2005
%
% Description:
% This function sets up an imaginary grid full of waypoints for UAV paths
%
% Inputs:
% InnerRadius km
% OuterRadius km
% StepRadius km
% StartRadius km
% StartRadius km
% ImageRequirement deg
% AngleWidth deg
% SAMlon deg
% SAMlat deg
% SAMalt m
% Height kft
% HeightSteps unitless
%
% Outputs:
% PixelLocations m
```

```

% Functions called:
% None

% Set HalfAngleWidth to be half of AngleWidth, converted to radians
HalfAngleWidth=(AngleWidth/2)*(pi/180);

% Set up waypoints at each degree of angle for every consecutive
% SAM ring until the max detection range.

r=InnerRadius;

while r<=StartRadius

for i=1:((AngleWidth+1)/2)

PixelLocations(i+(AngleWidth+1)*((r-InnerRadius)/StepRadius),1)...
=SAMlon-AngleDistance(r*sin(HalfAngleWidth-(i-1)*(pi/180)),...
SAMlat,2);

PixelLocations(i+(AngleWidth+1)*((r-InnerRadius)/StepRadius),2)...
=SAMlat-AngleDistance(r*cos(HalfAngleWidth-(i-1)*(pi/180)),...
SAMlat,1);

end

for i=(((AngleWidth+1)/2)+1):(AngleWidth+1)

PixelLocations(i+(AngleWidth+1)*((r-InnerRadius)/StepRadius),1)...
=2*SAMlon-...

PixelLocations((AngleWidth+2+(AngleWidth+1)*((r-InnerRadius)...
/StepRadius))-i,1);

PixelLocations(i+(AngleWidth+1)*((r-InnerRadius)/StepRadius),2)=...

PixelLocations(AngleWidth+2+(AngleWidth+1)*((r-InnerRadius)...
/StepRadius)-i,2);

end

PixelLocations(:,3)=SAMlat;

r=r+StepRadius;

end

% Solve for height at each waypoint
numppp=size(PixelLocations,1);

```

```

counter=numppp;
for i=1:HeightSteps
for x=1:numppp
PixelLocations(counter+1,1:2)=PixelLocations(counter+1-numppp,1:2);
PixelLocations(counter+1,3)=PixelLocations(x,3)+((InnerRadius+...
StepRadius*(floor(x/(AngleWidth+1.001))))*1000)*sin((asin(...
Height/StartRadius)/HeightSteps)*i);
counter=counter+1;
end
end

```

G.5 AngleDistance.m

```

function [Angle]=AngleDistance(Dist,Lat,x)
% AngleDistance
%
% 2Lt Daniel M Morales, USAF
% MIT Lincoln Laboratory
% 14 March 2005
%
% Description:
%
% This function uses WGS 84 data to give the angle length for any distance
% at a certain latitude.
%
% Inputs:
% Dist m
% Lat deg
% x 1 or 2
%
% Outputs:

```

% Angle deg

%

% Functions called:

% None

A= [0 110574 111319.2

1	110574.6	111302.4
2	110575.8	111252
3	110577.6	111168
4	110579.4	111050.4
5	110583	110898.6
6	110586.6	110713.8
7	110590.8	110495.4
8	110595.6	110243.4
9	110601.6	109957.8
10	110607.6	109639.2
11	110614.8	109287.6
12	110622	108902.4
13	110630.4	108484.8
14	110639.4	108034.2
15	110648.4	107550.6
16	110658.6	107034.6
17	110669.4	106485.6
18	110680.2	105904.8
19	110692.2	105292.2
20	110704.2	104647.2
21	110716.8	103970.4
22	110730	103262.4
23	110743.8	102522.6
24	110758.2	101751.6
25	110772.6	100950

26	110788.2	100117.8
27	110803.8	99255
28	110819.4	98361.6
29	110835.6	97438.8
30	110852.4	96486
31	110869.2	95504.4
32	110886.6	94493.4
33	110904.6	93453
34	110922.6	92385
35	110940.6	91288.2
36	110959.2	90163.8
37	110977.8	89011.8
38	110996.4	87832.2
39	111015.6	86626.2
40	111034.8	85393.8
41	111054	84135
42	111073.2	82851
43	111093	81541.2
44	111112.2	80206.2
45	111132	78846.6
46	111151.2	77463
47	111171	76056
48	111190.2	74625.6
49	111210	73171.8
50	111229.2	71695.8
51	111248.4	70197.6
52	111267.6	68677.8
53	111286.2	67137
54	111304.8	65575.8
55	111323.4	63994.2

56	111342	62392.8
57	111360	60772.2
58	111377.4	59133
59	111395.4	57475.2
60	111412.2	55800
61	111429	54107.4
62	111445.8	52398
63	111461.4	50673
64	111477.6	48931.8
65	111492.6	47175.6
66	111507.6	45405
67	111522	43620
68	111535.8	41821.8
69	111549	40010.4
70	111562.2	38186.4
71	111574.2	36351
72	111586.2	34504.2
73	111597.6	32646.6
74	111608.4	30779.4
75	111618.6	28902
76	111628.2	27015.6
77	111636.6	25121.4
78	111645	23219.4
79	111652.8	21309.6
80	111660	19393.2
81	111666.6	17471.4
82	111672	15543.6
83	111677.4	13611.6
84	111681.6	11674.8
85	111685.2	9734.4

```

86    111688.2    7791
87    111690.6    5845.8
88    111692.4    3898.2
89    111693.6    1949.4
90    111694.2    0 ];
if x==1
Angle=(Dist*1000)/(((Lat-fix(Lat))*(A(fix(Lat)+2,2)...
-A(fix(Lat)+1,2)))+A(fix(Lat)+1,2));
else
Angle=(Dist*1000)/(((Lat-fix(Lat))*(A(fix(Lat),3)...
-A(fix(Lat)+1,3)))+A(fix(Lat)+1,3));
end

```

G.6 gc2ecr.m

```

function ECR=gc2ecr(Lat,Lon,Alt)
% gc2ecr
%
% 2Lt Daniel M Morales, USAF
% MIT Lincoln Laboratory
% 14 March 2005
%
% Description:
%
% This function converts a geodetic state vector to ECR coordinates.
%
% The state vector must contain 3 elements (position), and this routine
% uses the supplied earth model.
%
% Inputs:
%
% Lat deg
% Lon deg

```

```

% Alt m
%
% Outputs:
% ECR m
%
% Functions called:
% deg2rad
% cos
% sin
%
% Reference: The majority of this code was translated into Matlab
% directly from MIT Lincoln Laboratory Java code for coordinate transforms.
ecc=0.081819301; % Earth's eccentricity
Re=6378.13649*1000; % Earth's radius, m
Lat=deg2rad(Lat);
Lon=deg2rad(Lon);
Clat=cos(Lat);
Slat=sin(Lat);
R=Re/sqrt(1-((ecc^2)*(Slat^2)));
ECR(1,1)=(R+Alt)*Clat*cos(Lon);
ECR(1,2)=(R+Alt)*Clat*sin(Lon);
ECR(1,3)=(R+Alt-((ecc^2)*R))*Slat;

```

G.7 CalculateDistances.m

```

function [Distances,WaypointTimes]=CalculateDistances(PixelLocations, ...
UAVSpeed)
% CalculateDistances
%
% 2Lt Daniel M Morales, USAF

```

```

% MIT Lincoln Laboratory
% 14 March 2005
%
% Description:
% This function calculates distances between all path waypoints
%
% Inputs:
% UAVSpeed m/s
% PixelLocations m
%
% Outputs:
% Distances m
% WaypointTimes s
%
% Functions called:
% None

% Calculate distances between each waypoint
for x=1:size(PixelLocations,1)
    for i=1:size(PixelLocations,1)
        Distances(x,i)=sqrt(((PixelLocations(i,1)-PixelLocations(x,1))...
            ^2)+((PixelLocations(i,2)-PixelLocations(x,2))^2)+((...
            PixelLocations(i,3)-PixelLocations(x,3))^2));
        if Distances(x,i)<0.001
            Distances(x,i)=0;
        end
        WaypointTimes(x,i)=Distances(x,i)/(UAVSpeed*1000);
    end
end

```

G.8 CreatePaths.m

```
function [Paths]=CreatePaths(AngleWidth,OuterRadius,InnerRadius,...  
StepRadius,HeightSteps,ImageRequirement)  
% CreatePaths  
%  
% 2Lt Daniel M Morales, USAF  
% MIT Lincoln Laboratory  
% 14 March 2005  
%  
% Description:  
% This function creates the UAV strategies, its paths.  
%  
% Inputs:  
% AngleWidth deg  
% OuterRadius km  
% InnerRadius km  
% StepRadius km  
% HeightSteps unitless  
% ImageRequirement deg  
%  
% Outputs:  
% Paths unitless  
%  
% Functions called:  
% vertcat  
tempRadius=(OuterRadius-InnerRadius)/StepRadius;  
% Enter SAM ring and leave ring at same level, all levels, all rings,  
% straight path  
for m=0:tempRadius  
for i=1:(HeightSteps+1)
```

```

for x=1:(tempRadius+2-m)
Paths1(i,x,m+1)=i*(tempRadius+2)*(AngleWidth+1)-(AngleWidth-...
((AngleWidth-5)/2+1))-(x-1)*(AngleWidth+1);
a=x;
end
for x=(a+1):(a+ImageRequirement)
Paths1(i,x,m+1)=Paths1(i,x-1,m+1)+1;
end
for x=(tempRadius+ImageRequirement+3-m):((tempRadius+...
ImageRequirement+2)+(tempRadius+1-2*m))
Paths1(i,x,m+1)=Paths1(i,x-1,m+1)+(AngleWidth+1);
end
end
end
Paths11=Paths1(:,:,1);
for x=1:(size(Paths1,3)-1)
Paths11=vertcat(Paths11,horzcat(Paths1(:,1:(min(find(Paths1...
(1,:,x+1)==0))-1),x+1),...
Paths1(:,(min(find(Paths1(1,:,x+1)==0))-1),x+1)*ones(1,size...
(Paths1(:,:,x+1),2)...-...
(min(find(Paths1(1,:,x+1)==0)-1))));...
end
% Enter SAM ring and leave ring at same level, all levels, all rings,
% zigzag path
for m=0:tempRadius
for i=1:(HeightSteps+1)
Paths2(i,1,m+1)=i*(tempRadius+2)*(AngleWidth+1)-(AngleWidth...
-((AngleWidth-5)/2+1));
for x=2:(tempRadius+2-m)
if mod(x,2)~=0

```

```

Paths2(i,x,m+1)=Paths2(i,x-1,m+1)-(AngleWidth+2);
else
    Paths2(i,x,m+1)=Paths2(i,x-1,m+1)-AngleWidth;
end
a=x;
end
for x=(a+1):(a+ImageRequirement)
    Paths2(i,x,m+1)=Paths2(i,x-1,m+1)+1;
end
for x=(tempRadius+ImageRequirement+3-m):((tempRadius+...
    ImageRequirement+2)+...
    (tempRadius+1-2*m))
if mod(x,2)~=0
    Paths2(i,x,m+1)=Paths2(i,x-1,m+1)+(AngleWidth+2);
else
    Paths2(i,x,m+1)=Paths2(i,x-1,m+1)+AngleWidth;
end
end
end
end
Paths22=Paths2(:,:,1);
for x=1:(size(Paths2,3)-1)
    Paths22=vertcat(Paths22,horzcat(Paths2(:,:,min(find(Paths2(:,:,x+1)...==0))-1),x+1),...
    Paths2(:,:,min(find(Paths2(:,:,x+1)==0))-1),x+1)*ones(1,size(...));
    Paths2(:,:,x+1),2)...-(min(find(Paths2(:,:,x+1)==0)-1))));;
end
% Enter SAM ring and leave ring at same level, all levels, all rings,
% block path

```

```

for m=0:tempRadius
for i=1:(HeightSteps+1)
j=0;
x=1;
Paths3(i,x,m+1)=i*(tempRadius+2)*(AngleWidth+1)-(AngleWidth...
-((AngleWidth-5)/2+1));
while Paths3(i,x,m+1)>((AngleWidth+1)*(m+1)+((i-1)*(tempRadius...
+2)*(AngleWidth+1)))
x=x+1;
if mod(x,2)~=0
Paths3(i,x,m+1)=Paths3(i,x-1,m+1)-(-1)^j;
else
Paths3(i,x,m+1)=Paths3(i,x-1,m+1)-(AngleWidth+1);
j=j+1;
end
a=x+1;
end
for x=a:(a+(ImageRequirement-1))
Paths3(i,x,m+1)=Paths3(i,x-1,m+1)+1;
end
j=0;
while Paths3(i,x,m+1)<(((tempRadius+1)*(AngleWidth+1)+1)+((i-1)*...
(tempRadius+2)*(AngleWidth+1)))
x=x+1;
if mod(x,2)~=0
Paths3(i,x,m+1)=Paths3(i,x-1,m+1)+(-1)^j;
else
Paths3(i,x,m+1)=Paths3(i,x-1,m+1)+(AngleWidth+1);
j=j+1;
end

```

```

end
end
end
Paths33=Paths3(:,:,1);
for x=1:(size(Paths3,3)-1)
Paths33=vertcat(Paths33,horzcat(Paths3(:,:,min(find(Paths3(:,:,...
x+1)==0))-1),x+1),...
Paths3(:,:,min(find(Paths3(:,:,x+1)==0))-1),x+1)*ones(1,size(......
Paths3(:,:,x+1),2)... ...
-(min(find(Paths3(:,:,x+1)==0)-1)))); ...
end
% Enter SAM ring high/low and leave low/high, all levels, all rings,
% straight path
for m=0:tempRadius
for i=1:fix(HeightSteps/2)
for x=1:(tempRadius+2-m)
Paths4(i,x,m+1)=i*(tempRadius+2)*(AngleWidth+1)-(AngleWidth-...
((AngleWidth-5)/2+1))-(x-1)*(AngleWidth+1);
a=x;
end
Paths4(i,a+1,m+1)=Paths4(i,a,m+1)+(AngleWidth+1)*(tempRadius+2)...
*(HeightSteps+1-i);
for x=(a+2):(a+1+ImageRequirement)
Paths4(i,x,m+1)=Paths4(i,x-1,m+1)+1;
end
for x=(tempRadius+ImageRequirement+4-m):((tempRadius+...
ImageRequirement+2)+(tempRadius+2-2*m))
Paths4(i,x,m+1)=Paths4(i,x-1,m+1)+(AngleWidth+1);
end
end

```

```

for i=(fix((HeightSteps)/2)+1):(HeightSteps+1)
for x=1:(tempRadius+2-m)
Paths4(i,x,m+1)=i*(tempRadius+2)*(AngleWidth+1)...
-(AngleWidth-((AngleWidth-5)/2+1))-(x-1)*(AngleWidth+1);
a=x;
end

Paths4(i,a+1,m+1)=Paths4(i,a,m+1)-(AngleWidth+1)*(tempRadius...
+2)*(i-1);
for x=(a+2):(a+2+ImageRequirement)
Paths4(i,x,m+1)=Paths4(i,x-1,m+1)+1;
q=x;
end

for x=q:((tempRadius+ImageRequirement+2)+(tempRadius+2-2*m))
Paths4(i,x,m+1)=Paths4(i,x-1,m+1)+(AngleWidth+1);
end
end
end

Paths44=Paths4(:,:,1);
for x=1:(size(Paths4,3)-1)
Paths44=vertcat(Paths44,horzcat(Paths4(:,:,min(find(Paths4(:,:,x+1)...
==0))-1),x+1),...
Paths4(:,:,min(find(Paths4(:,:,x+1)==0))-1),x+1)*ones(1,size(...));
Paths4(:,:,x+1),2)...-(min(find(Paths4(:,:,x+1)==0)-1)));
end

% Enter SAM ring high/low and leave low/high, all levels, all rings,
% zigzag path

for m=0:tempRadius
for i=1:fix(HeightSteps/2)
Paths5(i,1,m+1)=i*(tempRadius+2)*(AngleWidth+1)-(AngleWidth-...

```

```

((AngleWidth-5)/2+1));
for x=2:(tempRadius+2-m)
if mod(x,2)~ =0
Paths5(i,x,m+1)=Paths5(i,x-1,m+1)-(AngleWidth+2);
else
Paths5(i,x,m+1)=Paths5(i,x-1,m+1)-AngleWidth;
end
a=x;
end
Paths5(i,a+1,m+1)=Paths5(i,a,m+1)+(AngleWidth+1)*(tempRadius+2)...
*(HeightSteps+1-i);
for x=(a+2):(a+1+ImageRequirement)
Paths5(i,x,m+1)=Paths5(i,x-1,m+1)+1;
end
for x=(tempRadius+ImageRequirement+4-m):((tempRadius+...
ImageRequirement+2)+...
(tempRadius+2-2*m))
if mod(x,2)==0
Paths5(i,x,m+1)=Paths5(i,x-1,m+1)+(AngleWidth+2);
else
Paths5(i,x,m+1)=Paths5(i,x-1,m+1)+AngleWidth;
end
end
end
for i=(fix((HeightSteps)/2)+1):(HeightSteps+1)
Paths5(i,1,m+1)=i*(tempRadius+2)*(AngleWidth+1)-(AngleWidth-...
((AngleWidth-5)/2+1));
for x=2:(tempRadius+2-m)
if mod(x,2)~ =0
Paths5(i,x,m+1)=Paths5(i,x-1,m+1)-(AngleWidth+2);

```

```

else
    Paths5(i,x,m+1)=Paths5(i,x-1,m+1)-AngleWidth;
end
a=x;
end
Paths5(i,a+1,m+1)=Paths5(i,a,m+1)-(AngleWidth+1)*(tempRadius...
+2)*(i-1);
for x=(a+2):(a+2+ImageRequirement)
    Paths5(i,x,m+1)=Paths5(i,x-1,m+1)+1;
    q=x;
end
for x=q:((tempRadius+ImageRequirement+2)+(tempRadius+2-2*m))
if mod(x,2)==0
    Paths5(i,x,m+1)=Paths5(i,x-1,m+1)+(AngleWidth+2);
else
    Paths5(i,x,m+1)=Paths5(i,x-1,m+1)+AngleWidth;
end
end
end
end
Paths55=Paths5(:,:,1);
for x=1:(size(Paths5,3)-1)
    Paths55=vertcat(Paths55,horzcat(Paths5(:,:,min(find(Paths5(:,:,...
x+1)==0))-1),x+1,...));
    Paths5(:,:,min(find(Paths5(:,:,x+1)==0))-1),x+1)*ones(1,size(...));
    Paths5(:,:,x+1),2)...(-(min(find(Paths5(:,:,x+1)==0)-1))));;
end
% Enter SAM ring high/low and leave low/high, all levels, all rings,
% block path

```

```

for m=0:tempRadius
for i=1:fix(HeightSteps/2)
j=0;
x=1;
Paths6(i,x,m+1)=i*(tempRadius+2)*(AngleWidth+1)-(AngleWidth-...
((AngleWidth-5)/2+1));
while Paths6(i,x,m+1)>((AngleWidth+1)*(m+1)+((i-1)*(tempRadius...
+2)*(AngleWidth+1)))
x=x+1;
if mod(x,2)^=0
Paths6(i,x,m+1)=Paths6(i,x-1,m+1)-(-1)^j;
else
Paths6(i,x,m+1)=Paths6(i,x-1,m+1)-(AngleWidth+1);
j=j+1;
end
a=x+1;
end
Paths6(i,a,m+1)=Paths6(i,a-1,m+1)+(AngleWidth+1)*(tempRadius+2)...
*(HeightSteps+1-i);
for x=(a+1):(a+(ImageRequirement))
Paths6(i,x,m+1)=Paths6(i,x-1,m+1)+1;
end
while Paths6(i,x,m+1)<(((tempRadius+1)*(AngleWidth+1)+1)+((...
tempRadius+2)...
*(AngleWidth+1))*HeightSteps)
x=x+1;
if mod(x,2)==0
Paths6(i,x,m+1)=Paths6(i,x-1,m+1)+(-1)^j;
else
Paths6(i,x,m+1)=Paths6(i,x-1,m+1)+(AngleWidth+1);

```

```

j=j+1;
end
end
end

for i=(fix((HeightSteps)/2)+1):(HeightSteps+1)
j=0;
x=1;

Paths6(i,x,m+1)=i*(tempRadius+2)*(AngleWidth+1)-(AngleWidth-...
((AngleWidth-5)/2+1));
while Paths6(i,x,m+1)>((AngleWidth+1)*(m+1)+((i-1)*(tempRadius...
+2)*(AngleWidth+1)))
x=x+1;
if mod(x,2)~=0
Paths6(i,x,m+1)=Paths6(i,x-1,m+1)-(-1)^j;
else
Paths6(i,x,m+1)=Paths6(i,x-1,m+1)-(AngleWidth+1);
j=j+1;
end
a=x+1;
end

Paths6(i,a,m+1)=Paths6(i,a-1,m+1)-(AngleWidth+1)*(tempRadius+2)...
*(i-1);
for x=(a+1):(a+(ImageRequirement))
Paths6(i,x,m+1)=Paths6(i,x-1,m+1)+1;
end
j=0;
while Paths6(i,x,m+1)<((tempRadius+1)*(AngleWidth+1)+1)
x=x+1;
if mod(x,2)==0
Paths6(i,x,m+1)=Paths6(i,x-1,m+1)+(-1)^j;

```

```

else
    Paths6(i,x,m+1)=Paths6(i,x-1,m+1)+(AngleWidth+1);
    j=j+1;
end
end
end
end
Paths66=Paths6(:,:,1);
for x=1:(size(Paths6,3)-1)
    Paths66=vertcat(Paths66,horzcat(Paths6(:,:,min(find(Paths6(:,:,...
x+1)==0))-1),x+1),...
    Paths6(:,:,min(find(Paths6(:,:,x+1)==0))-1),x+1)*ones(1,size(... ...
    Paths6(:,:,x+1),2)-...
    (min(find(Paths6(:,:,x+1)==0)-1)))); ...
end
% Combine all paths into one matrix
Temp1=vertcat(vertcat(Paths44,Paths55),horzcat(vertcat(Paths11,Paths22),...
    vertcat(Paths11(:,end)... ...
    *ones(1,size(Paths44,2)-size(Paths11,2)),Paths22(:,end)*ones(1,size(... ...
    Paths44,2)-size(Paths22,2)))); ...
Temp2=vertcat(Paths33,horzcat(Temp1,Temp1(:,end)*ones(1,size(Paths33,2)... ...
    -size(Temp1,2)))); ...
Paths=vertcat(Paths66,horzcat(Temp2,Temp2(:,end)*ones(1,size(Paths66,2)... ...
    -size(Temp2,2)))); ...

```

G.9 Infeasible.m

```

function [Paths,CompletionTimes]=Infeasible(Paths,PixelLocations,... ...
WaypointTimes,Map,MapLegend,SAMlon,SAMlat,SAMalt)
% Infeasible

```

```

%
% 2Lt Daniel M Morales, USAF
% MIT Lincoln Laboratory
% 14 March 2005
%
% Description:
% This function eliminates UAV paths based on turn capabilities, UAV-SAM
% visibility, and terrain avoidance.
%
% Inputs:
% Paths unitless
% PixelLocations m
% WaypointTimes s
% Map m, DTED elevation data
% MapLegend DTED index variable
% SAMlon deg
% SAMlat deg
% SAMalt m
%
% Outputs:
% Paths unitless
% CompletionTimes s
%
% Functions called:
% ecr2gc
% los2
% DtedData
%
% Eliminates infeasible paths based on limitations of the UAV's turn
% capabilites. For this program, any turn smaller than about 90 degrees
% (actually pi/2-0.1) was automatically excluded as impossible.

```

```

s=1;
Rows=[];
for i=1:size(Paths,1)
for x=1:(size(Paths,2)-2)
A=[PixelLocations(Paths(i,x),1)-PixelLocations(Paths(i,x+1),1)...
PixelLocations(Paths(i,x),2)-PixelLocations(Paths(i,x+1)...
,2),PixelLocations(Paths(i,x),3)-PixelLocations(Paths(i,...
x+1),3)];
B=[PixelLocations(Paths(i,x+2),1)-PixelLocations(Paths(i,x+1),1)...
,PixelLocations(Paths(i,x+2),2)-PixelLocations(Paths(i,x+1)...
,2),PixelLocations(Paths(i,x+2),3)-PixelLocations(Paths(i,...
x+1),3)];
if acos(dot(A,B)/(norm(A)*norm(B)))<(pi/2-0.1)
Rows(s,1)=i;
s=s+1;
break;
end
end
end
Paths(Rows,:)=[];
for i=1:size(PixelLocations,1)
[PixelLocations(i,2),PixelLocations(i,1),PixelLocations(i,3)]=ecr2gc...
(PixelLocations(i,:));
end
% Eliminate infeasible paths based on whether the SAM is visible during all
% of the three points at which it is taking the image.
s=1;
Rows=[];
for i=1:size(Paths,1)
for x=4:size(Paths,2)

```

```

if (Paths(i,x)-Paths(i,x-3))==3
for j=(x-3):x
    Vis(j-x+4,1)=los2(Map,MapLegend,PixelLocations(Paths(i, ...
j),2),PixelLocations(Paths(i,j),1),SAMlat,SAMlon, ...
PixelLocations(Paths(i,j),3),SAMalt,'MSL','MSL');
end
if min(Vis)==0
    Rows(s,1)=i;
    s=s+1;
    break;
end
end
end
end
Paths(Rows,:)=[];
% Eliminate infeasible paths based on the UAV running into terrain.
s=1;
Rows=[];
for i=1:size(Paths,1)
    for x=1:size(Paths,2)
        [Alt,temp]=DtedData(PixelLocations(Paths(i,x),2),PixelLocations...
        (Paths(i,x),2),...
        PixelLocations(Paths(i,x),1),PixelLocations(Paths(i,x),1));
        if PixelLocations(Paths(i,x),3)<=Alt
            Rows(s,1)=i;
            s=s+1;
            break;
        end
    end
end

```

```

Paths(Paths,:)=[];
CompletionTimes(size(Paths,1),size(Paths,2))=0;
for x=1:size(Paths,1)
for i=2:size(Paths,2)
    CompletionTimes(x,i)=CompletionTimes(x,i-1)+WaypointTimes...
    (Paths(x,i),Paths(x,i-1));
end
end

```

G.10 Detect.m

```

function [Dep,Az,Detections,SNR,Locations1,Locations2,Locations3,RCS, ...
Range,RCSpPaths]=Detect(PixelLocations,Paths,UAVSpeed, ...
CompletionTimes,WaypointTimes,SAMecr,DetectionCutoff,DopplerCutoff)
% Detect
%
% 2Lt Daniel M Morales, USAF
% MIT Lincoln Laboratory
% 14 March 2005
%
% Description:
% This function takes UAV, path, and radar information and determines
% whether the UAV is detected or not throughout its path.
%
% Inputs:
% PixelLocations m, ECR coordinate system
% Paths unitless
% UAVSpeed m/s
% CompletionTimes s
% WaypointTimes s

```

```

% SAMecr m
% DetectionCutoff dB
% DopplerCutoff m/s
%
% Outputs:
% Dep deg
% Az deg
% Detections 1 (det) or 0 (not det)
% SNR dBsm
% Locations1/2/3 m
% RCS dBsm
% Range m
% RCSPaths dBsm
%
% Functions called:
% ecr2gc
% azimuth
% Change UAV Speed to m/s
UAVSpeed=UAVSpeed*1000; % m/sec
% Put together Azimuth / Depression angle look-up table for RCS
load RCS
% Calculate UAV location for all paths, updated every second
for i=1:size(Paths,1)
tempx=0;
tempy=0;
tempz=0;
t=1;
for x=2:size(Paths,2)
Delta=PixelLocations(Paths(i,x),:)-PixelLocations(Paths(i,x-1),:);
if x==2

```

```

Locations1(i,t)=PixelLocations(Paths(i,x-1),1)+((Delta(1,1)...
*UAVSpeed*t)/norm(Delta));
Locations2(i,t)=PixelLocations(Paths(i,x-1),2)+((Delta(1,2)...
*UAVSpeed*t)/norm(Delta));
Locations3(i,t)=PixelLocations(Paths(i,x-1),3)+((Delta(1,3)...
*UAVSpeed*t)/norm(Delta));
t=t+1;
tempt=t-2;
else
tempx=Delta(1,1)*(UAVSpeed*(t-CompletionTimes(i,x-1)))/norm...
(Delta);
tempy=Delta(1,2)*(UAVSpeed*(t-CompletionTimes(i,x-1)))/norm...
(Delta);
tempz=Delta(1,3)*(UAVSpeed*(t-CompletionTimes(i,x-1)))/norm...
(Delta);
Locations1(i,t)=PixelLocations(Paths(i,x-1),1)+tempx;
Locations2(i,t)=PixelLocations(Paths(i,x-1),2)+tempy;
Locations3(i,t)=PixelLocations(Paths(i,x-1),3)+tempz;
t=t+1;
tempt=t-1;
end
while t<CompletionTimes(i,x)
Locations1(i,t)=PixelLocations(Paths(i,x-1),1)+tempx+((...
Delta(1,1)*UAVSpeed*(t-tempt))/norm(Delta));
Locations2(i,t)=PixelLocations(Paths(i,x-1),2)+tempy+((...
Delta(1,2)*UAVSpeed*(t-tempt))/norm(Delta));
Locations3(i,t)=PixelLocations(Paths(i,x-1),3)+tempz+((...
Delta(1,3)*UAVSpeed*(t-tempt))/norm(Delta));
t=t+1;
end

```

```

end
end

for i=1:size(Locations1,1)
Locations1(i,(fix(CompletionTimes(i,end))+1):end)=Locations1(i,fix(...
CompletionTimes(i,end)));
Locations2(i,(fix(CompletionTimes(i,end))+1):end)=Locations2(i,fix(...
CompletionTimes(i,end)));
Locations3(i,(fix(CompletionTimes(i,end))+1):end)=Locations3(i,fix(...
CompletionTimes(i,end)));
end

% Calculate Azimuth / Depression angle for each path, updated every second
[SAMlat,SAMlon,SAMalt]=ecr2gc(SAMecr);
for i=1:size(Locations1,1)
for x=1:(size(Locations1,2)-1)
[Lat1,Lon1,Alt1]=ecr2gc([Locations1(i,x) Locations2(i, ...
x) Locations3(i,x)]);
[Lat2,Lon2,Alt2]=ecr2gc([Locations1(i,x+1) Locations2(i, ...
x+1) Locations3(i,x+1)]);
Range(i,x)=sqrt(((Locations1(i,x)-SAMecr(1))^2)+((Locations2(i, ...
x)-SAMecr(2))^2)+((Locations3(i,x)-SAMecr(3))^2));
Dep(i,x)=fix(rad2deg(asin((Alt1-SAMalt)/Range(i,x))));
if SAMlon<Lon1
Az(i,x)=fix(360+azimuth(Lat1,Lon1,Lat2,Lon2)-azimuth(Lat1...
,Lon1,SAMlat,SAMlon));
else
if azimuth(Lat1,Lon1,Lat2,Lon2)==0
Az(i,x)=fix(360-azimuth(Lat1,Lon1,SAMlat,SAMlon));
else
Az(i,x)=fix(azimuth(Lat1,Lon1,Lat2,Lon2)-azimuth(Lat1, ...
Lon1,SAMlat,SAMlon));
end
end
end

```

```

end
end
if Az(i,x)>360
Az(i,x)=Az(i,x)-360;
end
end
Dep=abs(Dep);
Az=abs(Az);
% Extract RCS values for every UAV location
for i=1:size(Dep,1)
for x=1:size(Dep,2)
if Az(i,x)==360
RCSpahs(i,x)=RCS(Dep(i,x)+1,Az(i,x)-1);
else
RCSpahs(i,x)=RCS(Dep(i,x)+1,Az(i,x)+1);
end
end
end
% Calculate SNR based on 10 dB SNR for 0 dBsm target at 120 km range
for i=1:size(RCSpahs,1)
for x=1:size(RCSpahs,2)
SNR(i,x)=log10((2.0736*(10^30))*((10^(RCSpahs(i,x)-4.5))/(
Range(i,x)^4)));
end
end
% Determine detections based on whether SNR is greater than or less than
% the specified detection cutoff, and disregard "detections" with Doppler
% velocity less than that specified cutoff.
for i=1:size(SNR,1)

```

```

for x=1:size(SNR,2)
if SNR(i,x)<DetectionCutoff
    Detections(i,x)=0; % not detected
else
    Detections(i,x)=1; % detected
end
if norm(UAVSpeed*cos(deg2rad(Az(i,x))))<DopplerCutoff % m/s
    Detections(i,x)=0; % not detected
end
end
end

```

G.11 FindPk.m

```

function [ProbKill]=FindPk(Detections,SNR,Locations1,Locations2,...
    Locations3,UAVSpeed,MissileSpeed,SAMecr,OuterRadius,InnerRadius,Az)
% FindPk
%
% 2Lt Daniel M Morales, USAF
% MIT Lincoln Laboratory
% 14 March 2005
%
% Description:
%
% This function takes the detections and determines whether the SAM can
% maintain a track on the UAV. From these calculations, it determines Pk.
%
% Inputs:
%
% Detections 1 (det) or 0 (not det)
%
% SNR dBsm
%
% Locations1/2/3 m

```

```

% UAVSpeed m/s
% MissileSpeed m/s
% SAMecr m
% OuterRadius km
% InnerRadius km
% Az deg
%
% Outputs:
% ProbKill from 0 to 1
%
% Functions called:
% None

% SAM Strategy 1: Shoot on first UAV detection - Probability of UAV being
% killed is proportional to amount of time UAV is in track.

for i=1:size(SNR,1)
if max(Detections(i,:))==0
ProbKill(i,1)=0.01;
%
% This number is not zero because it would excessively weight
% this component of the optimization
else
FirstInterceptTime=min(find(Detections(i,:)==1));
temp1=min(find(Detections(i,(FirstInterceptTime+1):end)==0));
while temp1<11
temp2=min(find(Detections(i,(FirstInterceptTime+temp1):...
end)==1));
if temp2
FirstInterceptTime=FirstInterceptTime+temp1+temp2-1;
temp1=min(find(Detections(i,FirstInterceptTime:end)==0));
else
break;

```

```

end
end

if min(find(Detections(i,(FirstInterceptTime+temp1):end)==1))
TimeUntilKill=fix(sqrt(((Locations1(i,FirstInterceptTime)...
-SAMecr(1))^2)+((Locations2(i,FirstInterceptTime)...
-SAMecr(2))^2)+((Locations3(i,FirstInterceptTime)-...
SAMecr(3))^2))/((UAVSpeed+MissileSpeed)*1000));
ProbKill(i,1)=sum(Detections(i,FirstInterceptTime:(...
FirstInterceptTime+TimeUntilKill)))/(TimeUntilKill+1);
if ProbKill(i,1)==1
ProbKill(i,1)=0.99;
% This number is not one because it would excessively
% weight this component
end
else
ProbKill(i,1)=0.01;
continue;
end
end
end

% SAM Strategy 2: Shoot when UAV is outbound - Probability of UAV being
% killed is proportional to amount of time UAV is in track.

for i=1:size(SNR,1)
for x=1:size(SNR,2)
if UAVSpeed*cos(deg2rad(Az(i,x)))<0
ShootTime=x;
break;
end
end
if(~exist('ShootTime'))

```

```

ShootTime=1000;
TimeUntilKill=1000;
else
TimeUntilKill=fix(sqrt(((Locations1(i,ShootTime)-SAMecr(1))^2)...
+((Locations2(i,ShootTime)...  

-SAMecr(2))^2)+((Locations3(i,ShootTime)-SAMecr(3))^2))/(...  

(UAVSpeed+MissileSpeed)*1000));
end
if (ShootTime+TimeUntilKill)>size(Detections,2)
ProbKill(i,2)=0.01;
else
ProbKill(i,2)=sum(Detections(i,ShootTime:(ShootTime+...  

TimeUntilKill)))/(TimeUntilKill+1);
end
if ProbKill(i,2)==1
ProbKill(i,2)=0.99;
% This number is not 1 because it would excessively weight  

% this component
elseif ProbKill(i,2)==0
ProbKill(i,2)=0.01;
% This number is not 0 because it would excessively weight  

% this component
end
end

```

G.12 ImageQuality.m

```

function [PictureQuality]=ImageQuality(CompletionTimes,Range,Paths,RCS,...  

Az,Dep)
% ImageQuality

```

```

%
% 2Lt Daniel M Morales, USAF
% MIT Lincoln Laboratory
% 14 March 2005
%
% Description:
% This function calculates image quality (NIIRS scale) for all paths.

%
% Inputs:
% CompletionTimes s
% Range m
% Paths unitless
% RCS dBsm
% Az deg
% Dep deg
%
% Outputs:
% PictureQuality 0 to 9
%
% Functions called:
% None

% Set up system of equations to solve for GIQE parameters
A = [1 (-1/log10((2.0736*(10^30))*((10^10)/(200000^4))));...
1 (-1/log10((2.0736*(10^30))*((10^10)/(20000^4))));];
b = [-(11.81+3.32*log10(cos(deg2rad(45))/200000)+(2*cos(deg2rad(abs...
(0-45))))-1);...
-(11.81+3.32*log10(cos(deg2rad(0))/20000)+(2*cos(deg2rad(abs...
(45-45))))-1)+6.5];
C = inv(A)*b;
%
% Use modified GIQE to find NIIRS quality of image taken

```

```

for i=1:size(Paths,1)
for x=1:size(Paths,2)
if (Paths(i,x+2)-2)==Paths(i,x)
ImagePoint(i,1)=round(CompletionTimes(i,x+1));
break;
end
end

PictureQuality(i,1)=11.81+3.32*log10(cos(deg2rad(min(abs(Az(i, ...
ImagePoint(i,1))-90),...
abs(Az(i,ImagePoint(i,1))-270))))/Range(i,ImagePoint(i,1)))...
+2*cos(deg2rad(... ...
abs(Dep(i,ImagePoint(i,1))-45))-1+C(1)-C(2)/log10((2.0736*...
(10^30))*...
(10^10)/(Range(i,ImagePoint(i,1))^4));
if PictureQuality(i,1)<0
PictureQuality(i,1)=0;
elseif PictureQuality(i,1)>9
PictureQuality(i,1)=9;
end
end

```

G.13 FindA.m

```

function [A]=FindA(PictureQuality,CompletionTimes,ProbKill)
% FindA
%
% 2Lt Daniel M Morales, USAF
% MIT Lincoln Laboratory
% 14 March 2005
%
```

```

% Description:
% This function finds the payoff matrix A.

%
% Inputs:
% PictureQuality 0 to 9
% CompletionTimes s
% ProbKill 0 to 1

%
% Outputs:
% A unitless

%
% Functions called:
% None

% Combine Pq and path completion times into temporary variable
temp=PictureQuality-0.001*CompletionTimes(:,end);

% Find inverse of Pk
ProbKillInv=1-ProbKill;

% Combine Pk and previous temp variable to get A
for i=1:size(PictureQuality,1)
    A(i,1)=temp(i,1)*ProbKillInv(i,1);
    A(i,2)=temp(i,1)*ProbKillInv(i,2);
    if Paths(fjndl)
        end

```

G.14 Nash.m

```

function [X,UAVPayoff,PathProbsX,PathProbsY,SAMPayoff,Y]=Nash(A)
% Nash

%
% 2Lt Daniel M Morales, USAF

```

```

% MIT Lincoln Laboratory
% 14 March 2005
%
% Description:
%
% This function uses the payoff matrix A to find the Nash equilibrium, or
% the best path for the UAV to take and the best strategy for the SAM to
% utilize against the UAV
%
% Inputs:
% A unitless
%
% Outputs:
% X unitless
% UAVPayoff unitless
% SAMPayoff unitless
% PathProbsX 0 to 1
% PathProbsY 0 to 1
% Y unitless
%
% Functions called:
% linprog
%
% Assign NumPaths and NumStrats to be the number of rows and columns in A
NumPaths=size(A,1);
NumStrats=size(A,2);
%
% Set up the costs, fx and fy, for the optimization routines
fx=ones(NumPaths,1);
fy=ones(NumStrats,1);
%
% Assign bx and by, part of the constraints
bx=ones(NumStrats,1);
by=ones(NumPaths,1);

```

```

% Call linprog to minimize fx
X=linprog(fx,-(A'),-bx,[],[],zeros(NumPaths,1),(10^5)*ones(NumPaths,1));
% Solve for UAV payoff
SumX=sum(X);
UAVPayoff=1/SumX;
% Solve for probabilities that UAV should choose each path
for i=1:size(X,1)
PathProbsX(i,1)=X(i,1)/SumX;
end
% Call linprog to minimize fy
Y=linprog(-fy,-(A),-by,[],[],zeros(NumStrats,1),(10^5)*ones(NumStrats,1));
% Solve for SAM payoff
Y=-Y;
SumY=sum(Y);
SAMPayoff=1/SumY;
% Sovle for probabilities that SAM should choose each strategy
for i=1:size(Y,1)
PathProbsY(i,1)=Y(i,1)/SumY;
end

```

G.15 Roll.m

```

function [BestPathX,BestStratY]=Roll(PathProbsX,PathProbsY)
% Roll
%
% 2Lt Daniel M Morales, USAF
% MIT Lincoln Laboratory
% 14 March 2005
%
% Description:

```

```

% This function generates random numbers, and essentially rolls the dice
% with predetermined probabilities to pick a UAV strategy
%
% Inputs:
% PathProbsX unitless
% PathProbsY unitless
%
% Outputs:
% BestPathX unitless
% BestStratY unitless
%
% Functions called:
% rand
%
% Generate random number to find UAV strategy
num=rand(1);
Tally=PathProbsX(1,1);
for i=1:size(PathProbsX,1)
if num<Tally
BestPathX=i;
break;
end
Tally=Tally+PathProbsX(i+1,1);
end
%
% Generate random number to find SAM strategy
num2=rand(1);
Tally2=PathProbsY(1,1);
for i=1:size(PathProbsY,1)
if num2<Tally2
BestStratY=i;
break;
end

```

```

end
Tally2=Tally2+PathProbsY(i+1,1);
end

```

G.16 ecr2gc.m

```

function [Lat,Lon,Alt]=ecr2gc(ECR)
% ecr2gc
%
% 2Lt Daniel M Morales, USAF
% MIT Lincoln Laboratory
% 14 March 2005
%
% Description:
% This function converts a ECR state vector to geodetic coordinates.
% The state vector must contain 3 elements (position), and this routine
% uses the supplied earth model.
%
% Inputs:
% ECR m
%
% Outputs:
% Lat deg
% Lon deg
% Alt m
%
% Functions called:
% rad2deg
%
% Reference: The majority of this code was translated into Matlab

```

```

% directly from MIT Lincoln Laboratory Java code for coordinate transforms.

ecc=0.081819301; % Earth's eccentricity

Re=6378.13649*1000; % Earth's radius, m

oneMinusEcc2=1-ecc^2;

x=ECR(1);

y=ECR(2);

z=ECR(3);

r=sqrt((x^2)+(y^2));

b=Re*sqrt(oneMinusEcc2);

% Compute longitude.

Lon=rad2deg(atan2(y,x));

l=(ecc^2)/2;

l2=l^2;

m=(r/Re)^2;

n=(oneMinusEcc2*z/b)^2;

i=-(2*l2+m+n)/2;

k=l2*(l2-m-n);

% double q = Math.pow((m + n - 4.0 * l2), 3.0) / 216.0 + m * n * l2;

qt=m+n-4*l2;

q=(qt^3)/216+m*n*l2;

D=sqrt((2*q-m*n*l2)*m*n*l2);

beta=(i/3)-(abs(q+D)^(1/3))-(abs(q-D)^(1/3));

sign=m-n;

if sign~=0

sign=sign/abs(sign);

end

t=sqrt(sqrt((beta^2)-k)-(beta+i)/2)-(sign*sqrt(abs(beta-i)/2));

r0=r/(t + l);

z0=oneMinusEcc2*z/(t-l);

% Compute latitude

```

```

Lat=rad2deg(atan(z0/(oneMinusEcc2*r0)));
% Compute altitude (m).

sign=t-1.0+l;
if sign~=0
sign=sign/abs(sign);
end

Alt=sign*sqrt(((r-r0)^2)+((z - z0)^2));

```

G.17 Visualize.m

```

function [Lat,Lon,Alt]=ecr2gc(ECR)
% ecr2gc

% 2Lt Daniel M Morales, USAF
% MIT Lincoln Laboratory
% 14 March 2005
%
% Description:
% This function converts a ECR state vector to geodetic coordinates.
% The state vector must contain 3 elements (position), and this routine
% uses the supplied earth model.
%
% Inputs:
% ECR m
%
% Outputs:
% Lat deg
% Lon deg
% Alt m
%
```

```

% Functions called:
% rad2deg
%
% Reference: The majority of this code was translated into Matlab
% directly from MIT Lincoln Laboratory Java code for coordinate transforms.

ecc=0.081819301; % Earth's eccentricity
Re=6378.13649*1000; % Earth's radius, m
oneMinusEcc2=1-ecc^2;
x=ECR(1);
y=ECR(2);
z=ECR(3);
r=sqrt((x^2)+(y^2));
b=Re*sqrt(oneMinusEcc2);

% Compute longitude.
Lon=rad2deg(atan2(y,x));
l=(ecc^2)/2;
l2=l^2;
m=(r/Re)^2;
n=(oneMinusEcc2*z/b)^2;
i=-(2*l2+m+n)/2;
k=l2*(l2-m-n);

% double q = Math.pow((m + n - 4.0 * l2), 3.0) / 216.0 + m * n * l2;
qt=m+n-4*l2;
q=(qt^3)/216+m*n*l2;
D=sqrt((2*q-m*n*l2)*m*n*l2);
beta=(i/3)-(abs(q+D)^(1/3))-(abs(q-D)^(1/3));
sign=m-n;
if sign~=0
sign=sign/abs(sign);
end

```

```

t=sqrt(sqrt((beta^2)-k)-(beta+i)/2)-(sign*sqrt(abs(beta-i)/2));
r0=r/(t + l);
z0=oneMinusEcc2*z/(t-l);
% Compute latitude
Lat=rad2deg(atan(z0/(oneMinusEcc2*r0)));
% Compute altitude (m).
sign=t-1.0+l;
if sign~=0
sign=sign/abs(sign);
end
Alt=sign*sqrt(((r-r0)^2)+((z - z0)^2));

```

G.18 create_uav.m

```

% create_uav
%
% 2Lt Daniel M Morales, USAF
% MIT Lincoln Laboratory
% 14 March 2005
%
% Description:
% This script is used to create a model of a UAV, similar to the Predator
% UAV with respect to size and shape
%
% Inputs:
% None
%
% Outputs:
% uav unitless
%
```

```

% Functions called:
% plate_collection
% frusta_collection

% Specify vertices and plates of both front and back wings on each side
left_front_wing_verts = [-1 0 4; -7.5 0 4.375; -7.5 0 5.125; -1 0 5.5];
left_front_wing_plates = [1 2 3; 1 3 4];
right_front_wing_verts = [1 0 4; 7.5 0 4.375; 7.5 0 5.125; 1 0 5.5];
right_front_wing_plates = [1 2 3; 1 3 4];
left_back_wing_verts = [-1 0 0.75; -3 -2 0.75; -3 -2 1.5; -1 0 1.5];
left_back_wing_plates = [1 2 3; 1 3 4];
right_back_wing_verts = [1 0 0.75; 3 -2 0.75; 3 -2 1.5; 1 0 1.5];
right_back_wing_plates = [1 2 3; 1 3 4];

% Collect plates together to form left and right wings, front and back
left_front_wing = plate_collection('verts',left_front_wing_verts, ...
'vert_inx',left_front_wing_plates,'one_sided',0);
right_front_wing = plate_collection('verts',right_front_wing_verts, ...
'vert_inx',right_front_wing_plates,'one_sided',0);
left_back_wing = plate_collection('verts',left_back_wing_verts, ...
'vert_inx',left_back_wing_plates,'one_sided',0);
right_back_wing = plate_collection('verts',right_back_wing_verts, ...
'vert_inx',right_back_wing_plates,'one_sided',0);

% Specify radius and length of cylinder segments
body_rad = [0.5 1 1 1.25 0];
body_len = [0 0.75 6.25 7.0 9];

% Form main body of UAV
body = frusta_collection('rad',body_rad,'len',body_len,'curve_flag',...
,'last');

% Assign 'uav' to include the body and all wings, as formed above
uav.components = {left_front_wing,right_front_wing,left_back_wing, ...
right_back_wing,body};

```

```
uav.name = 'uav';
```

G.19 loop3d.m

```
function RCSmean = loop3d(F)
% loop3d
%
% 2Lt Daniel M Morales, USAF
% MIT Lincoln Laboratory
% 14 March 2005
%
% Description:
% This function calculates average RCS values over all frequencies
%
% Inputs:
% F dBsm
%
% Outputs:
% RCSmean dBsm
%
% Functions called:
% None
%
% Loop through all frequencies to find RCS average values
Sum = 0;
for i=1:size(F,2)
    for x=1:size(F,3)
        for j=1:size(F,1)
            Sum = Sum + F(j,i,x);
        end
    RCSmean(i,x) = Sum/size(F,1);
```

```
Sum = 0;  
end  
end
```

G.20 matrix_interp.m

```
function [RCS]=matrix_interp(RCSb)  
% matrix_interp  
%  
% 2Lt Daniel M Morales, USAF  
% MIT Lincoln Laboratory  
% 14 March 2005  
%  
% Description:  
% This function switches rows and columns around in order necessary for  
% input into main program code.  
%  
% Inputs:  
% RCSb dBsm  
%  
% Outputs:  
% RCS dBsm  
%  
% Functions called:  
% interp  
% Sample RCS at twice the frequency  
for i=1:size(RCSb,1)  
temp(i,:) = interp(RCSb(i,:),2);  
end  
for i=1:size(temp,2)
```

```
RCS(:,i) = interp(temp(:,i),2);  
end
```